

Spektralanalys med hjälp av bildskärm

Bertil Ekenberg

Rapport LU/CS 75-3

Januari 1975



LUND UNIVERSITY
Department of Computer Sciences

1978



Lunds Universitet

Institutionen för informationsbehandling, särskilt numerisk analys

Sölvegatan 14A

223 62 Lund

Spektralanalys med hjälp av bildskärm

Bertil Ekenberg

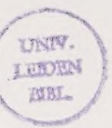
Rapport LU/CS 75-3

Januari 1975

Universitäts- und Landesbibliothek Bonn
Universitäts- und Landesbibliothek Bonn
Universitäts- und Landesbibliothek Bonn
Universitäts- und Landesbibliothek Bonn

Universitäts- und Landesbibliothek Bonn

Universitäts- und Landesbibliothek Bonn



Abstract

This report describes a program for spectral analysis using a graphic display unit. Measurements from spectral analysis are read by the program and the corresponding curve is drawn at the display screen. The curve is regarded as a sum of Gauss curves or Lorentz curves and possibly straight lines, and the problem is to determine parameters of these curves. By writing commands at the display console different curves may be drawn, and it is possible to compare these curves with the original curve. If the curves are close to each other, different optimization methods may be used to improve the fitting.

Innehåll

Inledning	1
Svårigheter vid kurvanpassningen	3
Manuell teknik för problemet	4
Datamaskinell bearbetning av problemet vid satsvis bearbetning	6
Datamaskinell bearbetning av problemet med hjälp av grafisk bildskärm	7
Beskrivning av ett spektralprogram, som utnyttjar grafisk bildskärm	9
Exempel på en körning	16
Sammanfattning	23
 Appendix 1: Lista över kommandon för spektralprogrammet	 A1
Appendix 2: Lista över variabler, som kan avläsas eller ändras från bildskärmen	A8
Appendix 3: Rättning av kommandon	A17
Appendix 4: Beskrivning av några numeriska metoder, som används i programmet	A18
Appendix 5: Styrkort för körning av programmet vid Lunds Datacentral	A27
Appendix 6: Beskrivning av en testgenerator	A29
 Referenser	

SPEKTRALANALYS MED HJÄLP AV BILDSKÄRM

Inledning

I samband med spektralanalys erhålles vissa mätvärden, som kan representeras i form av en kurva, t.ex.

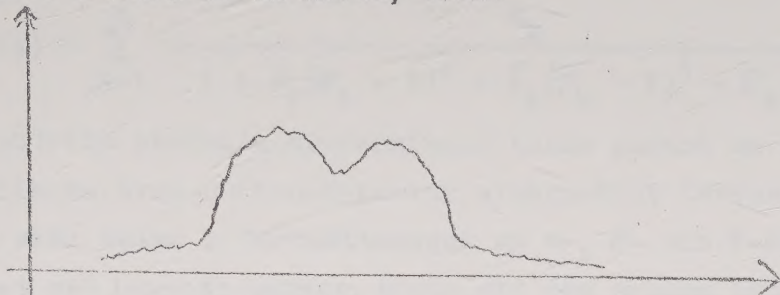


fig. 1

Horisontella axeln svarar mot olika energinivåer och vertikala axeln intensiteten vid de olika energinivåerna. Bortsett från vissa störningar antages kurvan kunna beskrivas som en summa av Gauss-kurvor, dvs vara en funktion av formen

$$f(t) = \sum_{k=1}^n \alpha_k e^{-\frac{(\beta_k - t)^2}{\gamma_k^2}} \quad (1)$$

I en funktion av slaget

$$g(t) = a e^{-\frac{(\beta - t)^2}{\gamma^2}} \quad (2)$$

svarar a mot höjden på kurvan

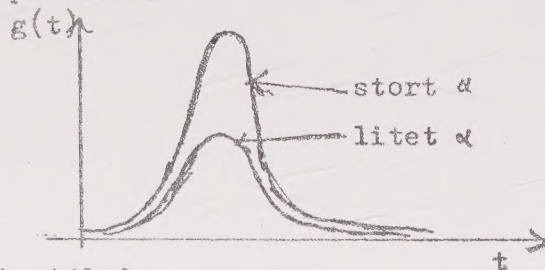


fig. 2

β mot toppens läge i sidled

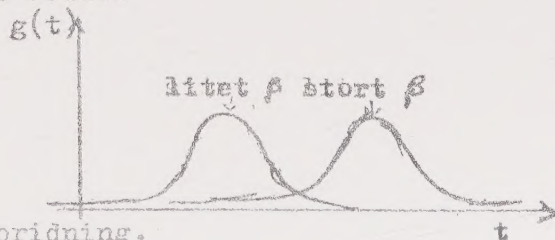


fig. 3

och γ mot toppens spridning.

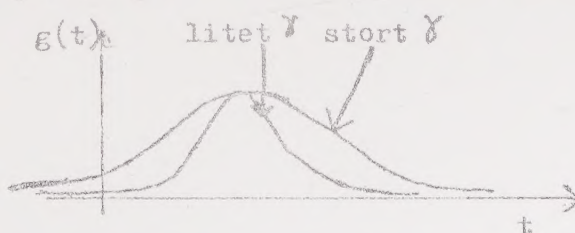
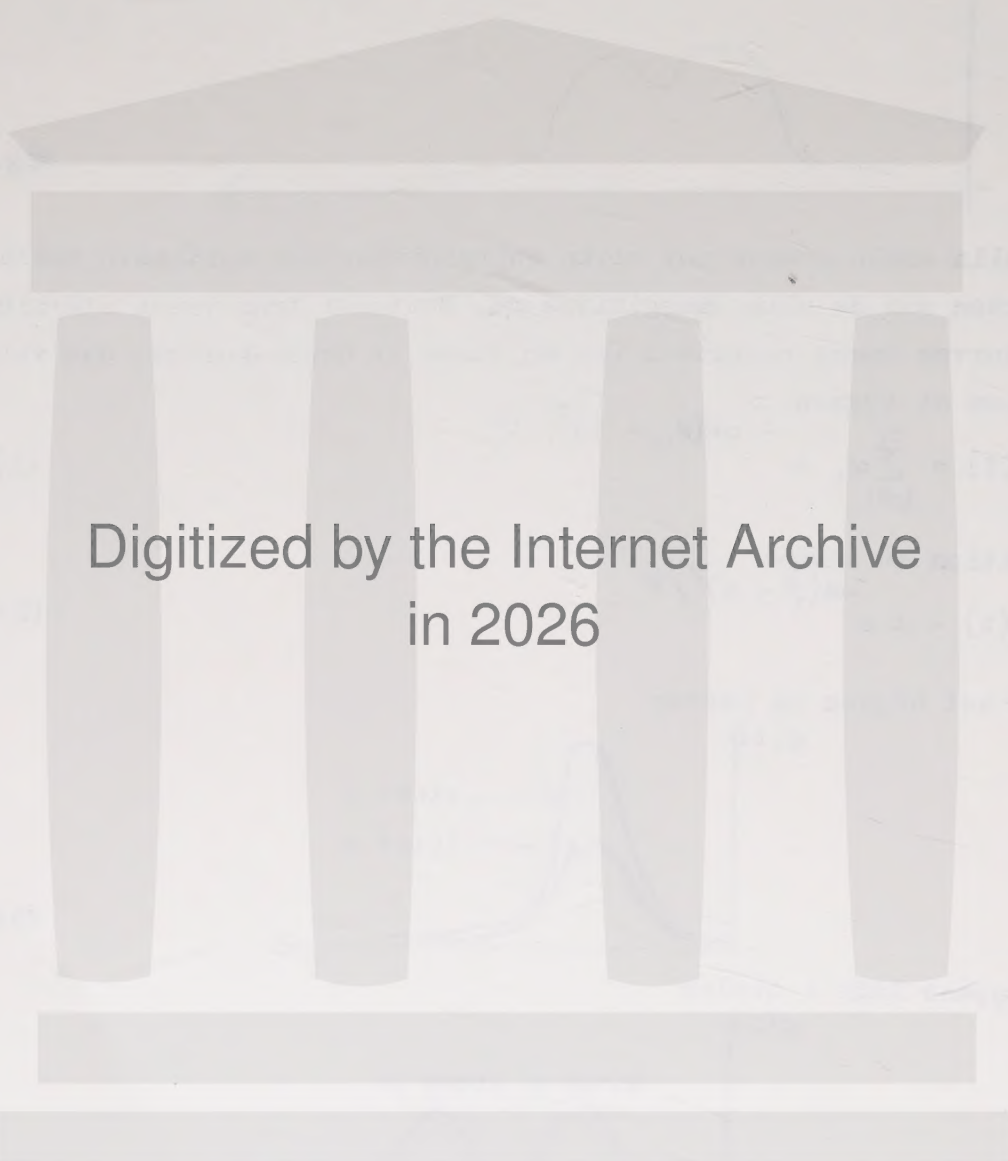


fig. 4

c är en konstant.



Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41553422_0032

Om man vill, kan man i stället antaga, att kurvan, bortsett från störningar, kan beskrivas som en summa av Lorentz-kurvor, dvs en funktion av formen

$$f(t) = \sum_{k=1}^n \frac{a_k}{1 + \gamma_k (\beta_k - t)^2 + \delta_k (\beta_k - t)^4 + \varepsilon_k (\beta_k - t)^6}$$

För att undvika störande upprepningar talas endast om Gauss-kurvor i fortsättningen även om Gauss-kurvor alternativt Lorentz-kurvor avses. Av samma skäl talas i fortsättningen om α -, β - och γ -variabler även för fallet med Lorentz-kurvor, trots att den senare kan varieras med α -, β -, γ -, δ - och ε -variabler.

Det prov man analyserar antages vara sammansatt av olika ämnen, vars spektra man ofta redan känner. Om man kan dela upp den ursprungliga kurvan i olika Gauss-kurvor, så kan man dra slutsatser om, vilka ämnen som ingår i provet, och i vilka proportioner ämnena ingår.

Man kan formulera problemet så att det gäller att anpassa en kurva som i figur 1 till en funktion som i formel (1), dvs bestämma antalet Gauss-kurvor kurvan är sammansatt av ($=n$), samt parametrarna α_k , β_k och γ_k till var och en av dessa Gauss-kurvor, så att felet vid anpassningen blir litet.

Svårigheter vid kurvanpassningen

Då man skall dela upp en kurva som i figur 1 i dess beståndsdelar, uppkommer vissa problem. För det första behöver en topp på kurvan inte nödvändigtvis motsvaras av exakt en Gauss-kurva. T.ex. kan en topp med utseendet



fig. 5

vara sammansatt av t.ex. två Gauss-kurvor.



fig. 6

Om man i detta fall antar, att man har att göra med en Gauss-kurva, blir anpassningen dålig.

Vidare störs mätvärdena av s.k. bakgrundsbrus på ett sätt, som man inte kan bortse från vid anpassningen. Om mätvärdena utan detta bakgrundsbrus skulle ge följande kurva

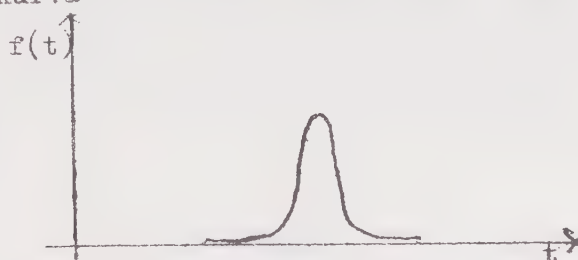


fig. 7

kan följande kurva ritas upp på ritutrustningen ansluten till mätapparaten

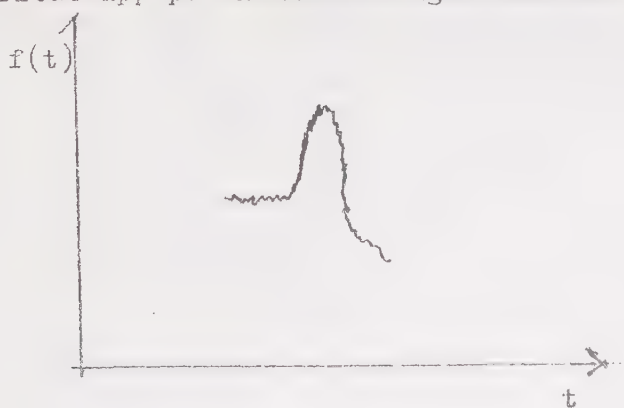


fig. 8

Manuell teknik för problemet

För närvarande används en manuell teknik vid spektralanalysen. Enligt Eva Andersson vid Kemiska institutionen vid Lunds Universitet kan man gå tillväga på följande sätt:

Mätningar på ett prov med delvis okända komponenter görs och resultatet ritas samtidigt upp på millimeterpapper.

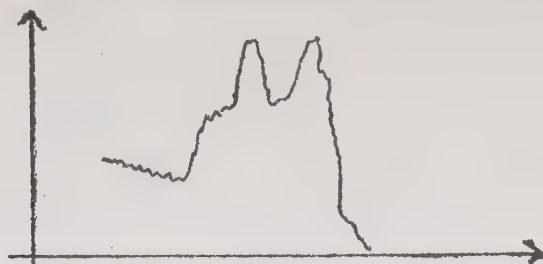


fig. 9

I det nedan beskrivna fallet kände man till en komponent, som ingick i provet, och därför jämfördes provet med det spektrum man fått med samma mätutrustning då enbart den kända komponenten ingick.

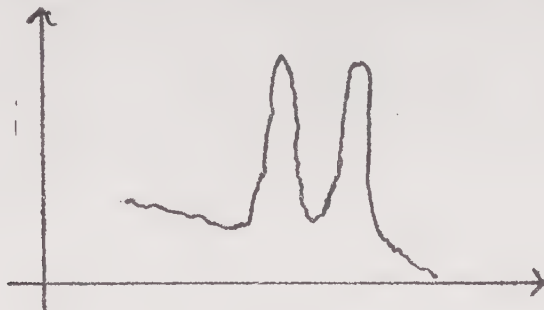


fig. 10

I de båda fallen antogs, att bakgrundskurvan var en rät linje, och att övriga störningar kunde försummas. I det intervall, där kurvan skilde sig väsentligt från bakgrundskurvan, beräknades differenserna mellan den ursprungliga kurvan och den rätta linjen, så att man fick fram en kurva, som antogs vara fri från bakgrundsstörningarna.

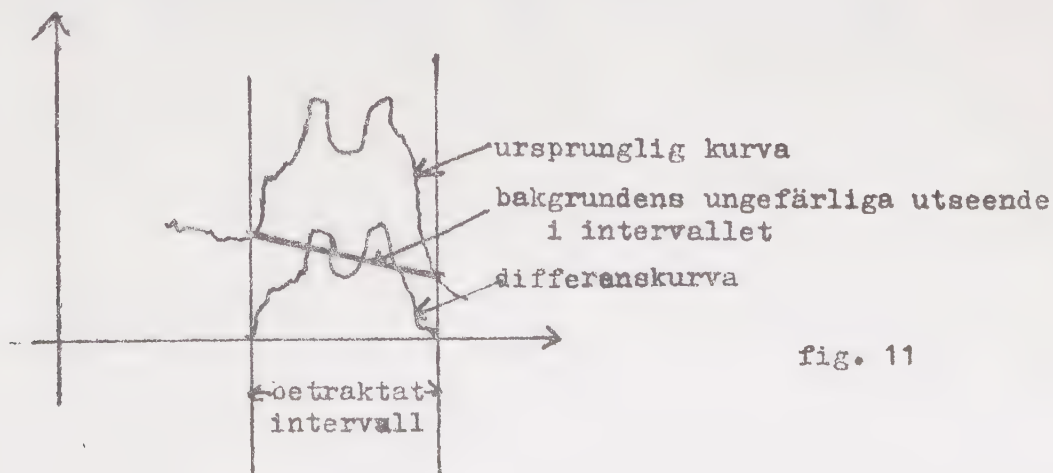


fig. 11

Kurvan med den rena ämnet (utan bakgrundskurvan) skalades därefter så att topparna sammanföll och så att kurvorna såg likadana ut vid ena ytterkanten (där man antog, att ingen annan Gauss-kurva störde). Härigenom kunde beräknas i vilken omfattning det kända ämnet ingick (motsvarande α_1 och α_2 i formel (1)). (Övriga parametrar var kända från början.) Differensen mellan kurvorna mättes därefter upp.



fig. 12

Om man vetat fler ämnen, som skulle ingå i provet, hade man fortsatt på samma sätt. Man antog därefter, att kurvan i vardera ytterkanten endast bestod av en topp, och beräknade hur var och en av dessa toppar fortsatte i det inre av intervallet. De framtagna topparnas kurvor subtraherades efter hand från kurvan. För varje sådan topp mättes med linjal toppens höjd, läge och halvvärdesbredd (motsvarande α_k , β_k resp. konstant γ_k i formel (1)).



fig. 13

De framtagna värdena jämfördes sedan med kända ämnens spektra, och på så sätt kunde de okända ämnena identifieras. Topp 1 och topp 3 i fig. 13 hörde i detta fall till ett ämne, topp 2 och topp 4 till ett annat ämne.

Datanaskinell bearbetning av problemet vid satsvis bearbetning

Vid Lunds Datacentral finns program, som kan tagas i bruk i samband med spektralanalys. Indata till programmet är mätvärden samt gissning på vilken kurva man har att göra med. Ur denna startapproximation försöker programmet beräkna en bättre approximation, och skriver ut resultat på radskrivare i form av tabeller och kurvor.

En nackdel med denna metod är, att det krävs en god startapproximation för att man ska få tillbaka ett förbättrat resultat. Om man t.ex. antagit, att en topp bestod av en Gauss-kurva i stället för två (eller tvärtom), ger metoden ett dåligt resultat. Man får då göra en ny gissning och köra om programmet tills man fått en lösning man är nöjd med. Om en kurva är svår att analysera är denna metod mycket tidsödande, beroende på att väntetiden mellan två körningar kan bli lång.

Datamaskinell bearbetning av problemet med hjälp av grafisk bildskärm

Vid Lunds Datacentral finns emellertid även en grafisk bildskärm. Denna kan användas så att man låter ett program rita upp startkurvor och approximerande kurvor samt parametervärden på bildskärmen varefter man omgående kan skriva en ny startapproximation på ett tangentbord anslutet till bildskärmen och begära en ny beräkning och få resultatet presenterat på bildskärmen praktiskt taget omgående. Med denna metod slipper man de långa väntetiderna mellan körningarna och kan lösa problemet betydligt snabbare, men metoden har även andra fördelar, vilka närmare framgår av ~~senare~~ **senare** **avsnitt**.

Bildskärmsutrustningen består av två större enheter, nämligen en konsol (Display Console) och en styrenhet (Display Controller). Konsolen består av själva bildskärmen, som är ett katodstrålerör och ser ut som bildrutan på en vanlig TV-apparat, samt ett tangentbord och en ljuspenna. Styrenheten kan närmast liknas vid en liten datamaskin, eftersom den är försedd med ett eget minne, i vilket såväl instruktioner som bildbeskrivningar lagras. Den är vidare försedd med ett register, där t ex additioner av heltal och skiftoperationer kan utföras. Styrenheten för bildskärmen är ansluten via en kommunikationsväxel till centralenheten för huvudmaskinen UNIVAC 1108, så att man kan överföra program och data mellan de olika enheterna.

Styrenheten för bildskärmen är försedd med ett minne med 16K ord, dvs 16 384 ord, som vart och ett består av 18 bitar. I ett ord kan man t ex ha en instruktion av typen hämta ett ord från minnet och placera det i ett register, lagra registerinnehållet i minnet, addera ett tal till registret, hoppa till en annan instruktion, hoppa över nästa instruktion om registret innehåller talet 0 etc. Operationstiderna varierar mellan 0.7 och 2.1 mikrosekunder (10^{-6} sekunder), beroende på vilken operation som utförs. Minnescykeltiden är 0.7 mikrosekunder.

Bland instruktionerna finns också operationer för att bestämma om man vill ha avbrott från någon i enheten inbyggd klocka eller avbrott om någon tangent vid konsolen trycks ned etc. Det finns bl.a. en klocka, som ger avbrott 60 gånger i sekunden och som normalt används för att starta en rutin, som i sin tur startar bildgenereringen. Bilden på bildskärmen måste nämligen genereras på nytt ca 60 gånger per sekund, eftersom den bara ligger kvar på skärmen en kort tid.

Konsolen är försedd med en bildskärm, ett tangentbord och en ljuspenna. Bildskärmen ser ut som bildrutan på en TV. Storleken på bildytan är 30 x 30 cm, och bildytan har 1024 x 1024 adresserbara punkter. En bild genereras i princip som ett antal rätta linjer. Enheten läser från styrenhetens minne instruktion för instruktion parallellt med styrenhetens övriga arbete så snart bildgenereringen startats av ett program i styrenheten.

En instruktion kan ha innebörden: rita en rät linje till punkten med koordinaterna (300, 400). Bildskärmen tänks ha ett x-y-koordinatsystem med x-axeln riktad åt höger och y-axeln riktad uppåt med $x=0$, $y=0$ i nedre vänstra hörnet. När instruktionen påträffas, förflyttas katodstrålerörets elektronstråle med konstant hastighet från den punkt, där elektronstrålen för ögonblicket träffar skärmen, till punkten med koordinaterna $x=300$, $y=400$. Förutom möjligheten att dra en rät linje till en bestämd punkt, kan man förflytta strålen genom att ange ökningen eller minskningen i x- och y-led. Det finns tre olika ljusstyrkor: hög intensitet, låg intensitet och helt släckt linje (blank linje). Man kan ange om en linje skall vara heldragen, eller streckad eller om endast en punkt skall ritas ut. Det finns också instruktioner av typen: Skriv bokstaven A på bildskärmen, och en teckengenerator, som kan generera bokstäverna A-Z och siffrorna 0-9 och en del specialtecken, t ex en del grekiska bokstäver, integraltecken, rottecken och procenttecken.

Tangentbordet ser ut som tangentbordet på en vanlig skrivmaskin, men har fler tangenter. Om man i programmet givit instruktion om att avbrott skall accepteras, får man ett avbrott varje gång man trycker på en tangent. Då avbryts den programsekvens, som just exekveras, och ett hopp sker till en fix minnesadress. I denna minnescell skall man tidigare ha placerat en hoppinstruktion till ett programavsnitt som analyserar avbrottsorsaken. Med en särskild instruktion kan man få reda på t ex att avbrottet kom från tangentbordet och att det var tangenten A som trycktes ned. Programmeraren kan sedan använda denna information hur han själv vill.

Ljuspennan ser ut på följande sätt:

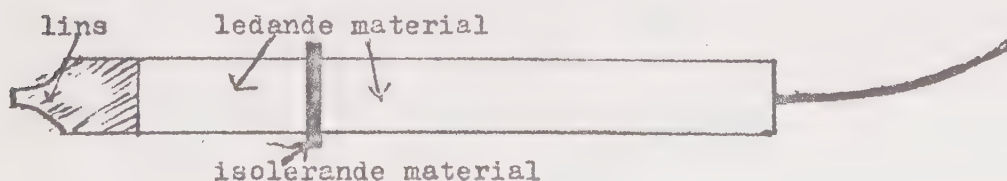


fig. 14

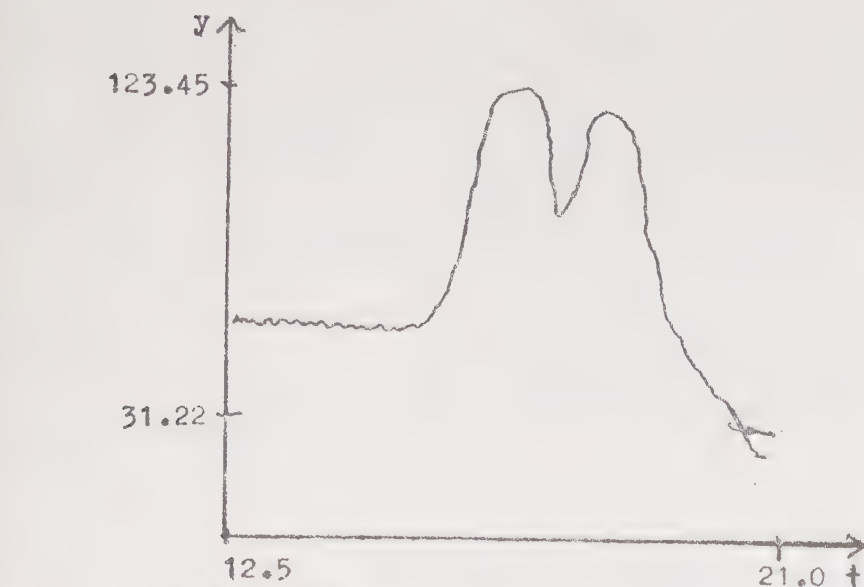
Om man håller i bakre delen av ljuspennan och sätter pekfingeret på främre delen av pennan, sluts en strömkrets och, om linsen är riktad mot en ljus punkt på skärmen och programmet tillåter avbrott, genereras ett avbrott. Man kan då avläsa var i listan över instruktioner för konsolen, som avbrottet skedde. Liksom vid avbrott från tangentbordet är det här programmerarens sak att avgöra hur informationen skall användas.

Beskrivning av ett spektralprogram, som utnyttjar grafisk bildskärm.

Programmet är, grovt sett, uppdelat i en del, som finns i bildskärms-
utrustningens minne, och en del som finns i huvudmaskinens minne. Delarna
innehåller rutiner för kommunikation mellan delarna, så att man kan sända
data, representerat som en följd heltal, från bildskärmen till huvudmaskinen
och tvärtom. Vid dessa sändningar är första heltalet en kod, som talar om,
vilken beräkning som skall utföras.

När man skall använda programmet måste man först göra en initiering, som
består i, att man gör bildskärmen beredd att ta emot program, överför
bildskärmsdelen av programmet från skrivminne via huvudmaskinen till bildskärmens
minne, startar denna programdel, så att de efterföljande sändningarna från
huvudmaskinen kan tas emot och tolkas på rätt sätt. Därefter startas
programmet i huvudmaskinen, vilket börjar med att läsa in denmätvärden man
vill analysera. Hittills har inläsningen skett från kortläsare, men
programmet är förberett även för andra typer av inmatning. Ur mätvärdena
beräknas motsvarande heltalskoordinater för bildskärmens 1024 x 1024 -punkters
rutnät, och dessa koordinater, samt uppgifter om koordinataxlarnas läge
och kurvans största och minsta värde, sänds till bildskärmsenheten. Här
beräknas ur koordinaterna motsvarande bildgenererande instruktioner, och
man får en bild som kan se ut t.ex. så här

BILD 1 A



$$y = \sum \alpha_k e^{-(\beta_k - t)^2 \gamma_k}$$

$\alpha_1=0$	$\alpha_2=0$	$\alpha_3=0$	$\alpha_4=0$	$\alpha_5=0$
$\beta_1=0$	$\beta_2=0$	$\beta_3=0$	$\beta_4=0$	$\beta_5=0$
$\gamma_1=0$	$\gamma_2=0$	$\gamma_3=0$	$\gamma_4=0$	$\gamma_5=0$

Talen till vänster om y-axeln svarar mot största och minsta funktionsvärdet för mätvärdena, x-axeln mot $y=0$. Talen under x-axeln svarar mot minsta och största x-värde för mätvärdena, y-axeln mot $x=$ minsta x-värdet. Om man vill kan man redan vid inläsningen av mätvärdena ange x-värdena på ett sätt man själv finner lämpligt. Om t ex de verkliga mätvärdena svarar mot intervallet (1012.5, 1021.0), kan man i stället räkna med intervallet (12.5, 21.0), eftersom man då kan få resultat utskrivna med större noggrannhet och vidare slipper skriva inledande siffror för x-värdena, när man sitter vid bildskärmen.

Under bilden anges, vilken funktion som kommer att användas, om man begär att en kurva ska ritas upp på bildskärmen. Man kan välja mellan 8 funktionsformer. Man har dessutom möjligheten att endast sända α -, β -, γ -, δ - och ϵ -värdena.

$$1. \quad y = \sum_k \alpha_k e^{-\frac{(\beta_k - t)^2}{\gamma_k}} \quad (3.1)$$

$$2. \quad y = \sum_k \alpha_k e^{-\frac{(\beta_k - t)^2}{\gamma_k^2}} \quad (3.2)$$

$$3. \quad y = \sum_k \alpha_k e^{-\frac{(\beta_k - t)^2}{(2\gamma_k^2)}} \quad (3.3)$$

$$4. \quad y = \sum_k \alpha_k e^{-c \frac{(\beta_k - t)^2}{\gamma_k^2}} \quad (3.4)$$

$$5. \quad y = \sum_{k=3}^n \alpha_k e^{-c \frac{(\beta_k - t)^2}{\gamma_k^2}} + \text{den rätta linje som går genom } (\alpha_1, \beta_1) \text{ och } (\alpha_2, \beta_2) \quad (3.5)$$

$$6. \quad y = \text{det polygontåg, som går genom } (\alpha_1, \beta_1), (\alpha_2, \beta_2), \dots, (\alpha_n, \beta_n) \quad (3.6)$$

$$7. \quad \text{Endast } \alpha\text{-, } \beta\text{-, } \gamma\text{-, } \delta\text{- och } \epsilon\text{-värden sänds} \quad (3.7)$$

$$8. \quad y = \sum_k \frac{\alpha_k}{1 + \gamma_k (\beta_k - t)^2 + \delta_k (\beta_k - t)^4 + \epsilon_k (\beta_k - t)^6} \quad (3.8)$$

$$9. \quad y = \sum_{k=3}^n \frac{\alpha_k}{1 + \gamma_k (\beta_k - t)^2 + \delta_k (\beta_k - t)^4 + \epsilon_k (\beta_k - t)^6} + \text{den rätta linje som går genom } (\alpha_1, \beta_1) \text{ och } (\alpha_2, \beta_2) \quad (3.9)$$

I fallen 4 och 5 är c en konstant, som kan bestämmas av användaren (genom ett kommando $C=\text{tal}$). C har startvärdet $4 \cdot \ln(2)$, vilket gör att γ i formel (2) svarar mot halvvärdesbredden.

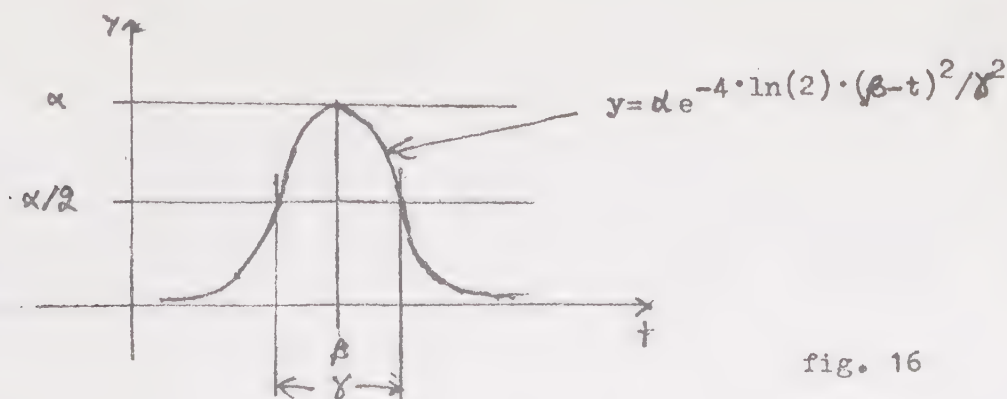


fig. 16

Formerna 2 och 3 är specialfall av form 4 ($\theta=1$ resp. 0.5). I fall 6 fås den rätta linjen genom (α_1, β_1) , (α_2, β_2) då $x < \alpha_2$, linjen genom (α_2, β_2) , (α_3, β_3) då $\alpha_2 < x < \alpha_3$ o.s.v. så länge $\alpha_k < \alpha_{k+1}$. Speciellt måste α_1 vara mindre än α_2 . Det är tillåtet att ange punkter utanför det intervall, som visas på skärmen.

Man bestämmer vilken funktionsform programmet skall använda genom att ge en variabel FX motsvarande värde. Om man t.ex. vill arbeta med funktionen (3.4) skriver man kommandot FX=4 vid bildskärmen, varvid texten under kurvan ändras. Nästa gång man begär att få uppritad den kurva, som svarar mot den funktion, som står på bildskärmen, och de värden på $\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2, \dots, \alpha_n, \beta_n, \gamma_n$, som man skrivit vid skärmen, överförs dessa värden till huvudmaskinen, och motsvarande kurva beräknas och ritas upp på bildskärmen. Man ändrar värde på α -, β - och γ -variablerna genom kommandon av typen $\alpha 1=97$, $\beta 1=12.789$, $\gamma 3=1.5$ etc. och begär att få motsvarande kurva beräknad genom kommandot NY KURVA. (Den allmänna formen för kommandona ges i appendix 1.)

I övre vänstra hörnet på bilden står numret och versionen på bilden. Varje gång man får en ny kurva uppritad, ökas numret med ett. Denna version kallas version A. Om man t.ex. begär att få en ny kurva uppritad, kallas den bild man då har version B. De senaste bilderna sparas, och man kan få dem uppritade igen genom kommandon av typen BILD 1B, BILD 2 etc. (Om versionsnummer saknas väljs den senast sparade versionen.) På detta sätt kan man t.ex. jämföra den senast uträknade kurvan med tidigare kurvor och eventuellt återgå till en tidigare approximation med dess parametervärden.

Man kan nu börja försöka bestämma vilken kurva man har att göra med. För att illustrera hur detta kan gå till, kan vi anta, att vi skall arbeta med kurvan i fig. 15. Vi börjar med att försöka dra bort bakgrundsbruset. Vi provar med den rätta linjen genom (15,60) och (20,50) och kan då skriva följande kommandon:

FX=6

$\alpha 1=15$

$\beta 1=60$

$\alpha 2=20$

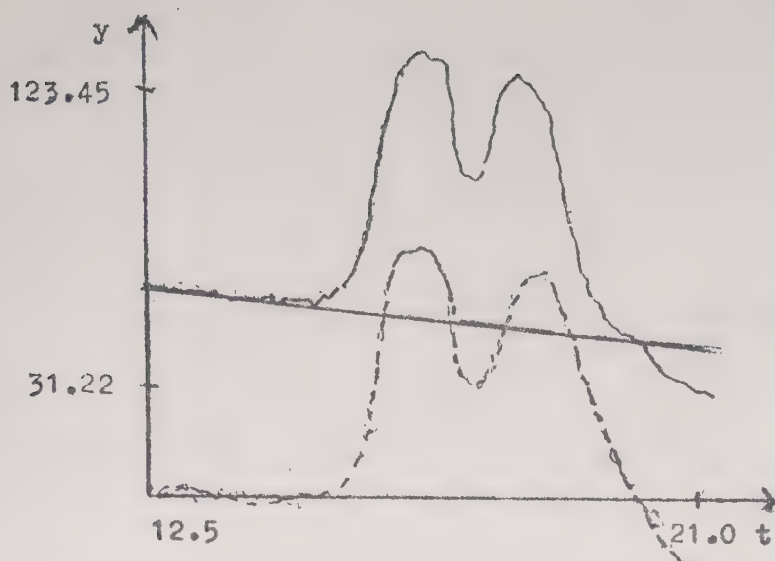
$\beta 2=50$

NY KURVA.

Hur kommandon skrivs är närmare beskrivet i appendix 3.

Efter det senaste kommandot sändes värdena på FX och α -, β - och γ -variablerna till huvudmaskinen, som beräknar hur denna kurva ser ut, och resultatet sändes tillbaka till bildskärmen. Vi kan nu ha följande bild framför oss.

BILD 2 A



POLYGONTÅG GENOM $(\alpha_1, \beta_1), \dots, (\alpha_n, \beta_n)$, MED $\alpha_n > \alpha_{n+1}$

$\alpha_1=15$	$\alpha_2=20$	$\alpha_3=0$	$\alpha_4=0$	$\alpha_5=0$
$\beta_1=60$	$\beta_2=50$	$\beta_3=0$	$\beta_4=0$	$\beta_5=0$

fig. 17

Vi har nu framför oss tre kurvor, nämligen den kurva vi skall approximera, vilken uppritas svagt (låg intensitet), den kurva vi begärt att få uppritad, vilken uppritas starkt (hög intensitet), och differensen mellan kurvorna (streckade linjer, hög intensitet).

Vi kan här t.ex. försöka approximera bakgrunden bättre genom låta linjen luta mera (t.ex. kommandona $\beta_2=5$, NY KURVA skrivs) eller genom att approximera bakgrunden med olika räta linjer i olika intervall (t.ex. genom kommandona $\alpha_3=21, \beta_3=31.22$, NY KURVA skrivs).

Låt oss anta, att vi är nöjda med den approximation vi gjort och vill använda differenskurvan vid den fortsatta approximationen. Vi kan då skriva kommandot

DIFF

vilket gör, att den ursprungliga kurvan ersätts med differenskurvan.

Vi får då bilden som i fig. 18. Den nya bildens utseende har beräknats, så att kurvan på nytt skall fylla ut det område, som reserverats för kurvan på skärmen. Den nya skalan på kurvan framgår av värdena på y-axeln.

BILD 3 A

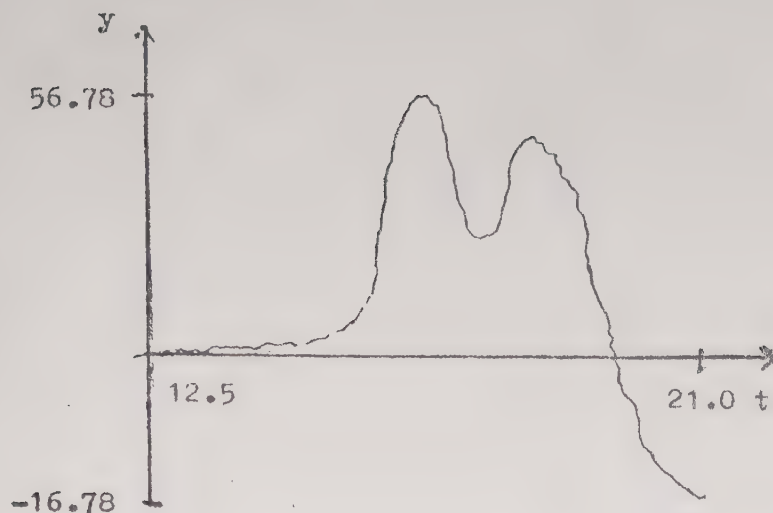


fig. 18

Eftersom vi nu endast är intresserade av topparna på kurvan, vill vi bortse från de delar av kurvan, som är längst till vänster och längst till höger på figuren. Vi anger därför vilka punkter vi vill arbeta vidare med genom att ange numret på den minsta resp. störstapunkten som vi skall arbeta med i fortsättningen. Om vi t.ex. vill bortse från de 20 första punkterna (längst till vänster på bilden) skriver vi kommandot

MININD=21

MININD anger minsta index man betraktar av den inlästa kurva, MAXIND på motsvarande sätt maximala index. Om man har svårt att direkt avgöra, vilket minsta resp. största index man vill betrakta är, kan man pröva med olika värden på MININD och MAXIND. Men varje gång MININD eller MAXIND ändras, görs en ny uträkning av koordinaterna för den kurva man vill anpassa, eventuellt anpassande kurva och läget på koordinataxlarna. Den är därför bättre att peka med ljuspennan på första eller sista punkten man vill betrakta på någon av de kurvor man har uppritad framför sig, varvid motsvarande punkts index skrivs ut nedtill på skärmen. I detta sammanhang är det bäst att peka på någon kurva som ritas med hög intensitet (anpassande kurvan tex.), eftersom en linje som ritas med låg intensitet inte alltid ger avbrott. Om endast en sådan kurva finns på skärmen, kan man behöva gå tillbaka till en tidigare bild, där flera kurvor är uppritade, när man ska bestämma index.

Fördelarna med att begränsa intervallet är dels, att endast de punkter man är intresserad av beräknas, vilket sparar maskintid och ger en bättre bild, dels att anpassningar med numeriska metoder endast tar hänsyn till felet i det intervall man vill betrakta och inte störs av bakgrundsbrus och andra toppar på den kurva man betraktar. Om man har en kurva med flera toppar klart skilda från varandra, kan man alltså anpassa en topp i taget.

Efter att vi ändrat värdena på MININD och MAXIND kan vi t.ex. ha följande kurva på skärmen.

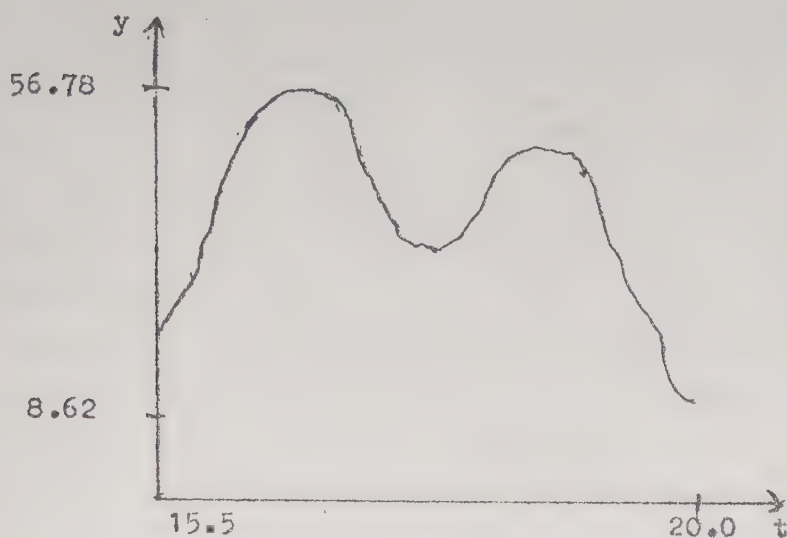


fig.19

Vi kan nu anta, att denna kurva approximativt är en summa av två Gauss-kurvor, dvs $n=2$ i formel (1). Vi ändrar värdet på FX till 4 (formel 3.4) och gissarmotsvarande parametervärden, t.ex.

FX=4

$\alpha_1=50.$

$\beta_1=16.7$

$\gamma_1=1.5$

$\alpha_2=45$

$\beta_2=18.3$

$\gamma_2=1$

Om α_3, α_4 o.s.v. inte redan haft värdet 0, hade vi behövt ge dessa variabler värdet 0. Vi kan nu begära att få denna kurva uppritad genom kommandot

NY KURVA

Om kurvorna på skärmen inte avviker för mycket från varandra, kan man försöka optimera parametrarna med någon numerisk metod genom att skriva t.ex.

GAUSS

eller

SIMPLEX

eller

STEPIT

I första fallet görs ett försök till approximation med Gauss-Newton's metod.

I andra fallet görs anpassning med den metod Nelder-Mead (1965) kallar

Simplex-metoden, och i tredje fallet anropas rutinen Stepit, som skrivits

av Chandler (1965). Metoderna är i denna rapport beskrivna i appendix 4.

I samtliga dessa fall används som startapproximation de värden på FX , α , β och γ , som användes förra gången en kurva beräknades. Man kan också genom ett kommando

TEST

styra valet av approximationsmetod, så att ett test får avgöra om Gauss-approximation eller Simplex-approximation skall användas. Därefter utvalda approximationsmetoden används därefter vid anpassningen.

Genom att onväxlande använda olika numeriska metoder och gissningar av α -, β - och γ -värden och få motsvarande kurvor uppritade, bör man kunna bestämma vilken kurva man har att göra med.

Om man bestämt parametrarna för några toppar, men inte alla, med tillräcklig noggrannhet, kan man ta bort dessa toppar med kommandot DIFF. När man är helt klar med körningen, avbryter man förbindelsen mellan bildskärmsenheten och huvudmaskinen genom kommandot

SLUTA

varefter man kan stänga av bildskärmsenheten.

Det finns ytterligare några kommandon beskrivna i appendix 1 och appendix 2.

Exempel på en körning

Här ges ytterligare ett exempel på, hur en kurva kan analyseras. I detta exempel framställdes kurvan med hjälp av ett program, som genererade testdata (se appendix 6). Kurvan skulle ha mellan 50 och 75 mätvärden med början med $t=0.0$ och steglängden 0.1. Den skulle ha två toppar och eventuellt en rät linje (närmare bestämt, FX skulle ha värdet 4 med sannolikheten 0.7 och 5 med sannolikheten 0.3). För varje topp gällde, att α -värdet var mellan 50 och 100, β -värdet mellan 1 och 4 och γ -värdet mellan 0.5 och 3. Om en rät linje lades till, skulle den för $t=0$ och $t=5$ ligga mellan -2 och 2. Vidare fick varje punkt ha en störning mellan -1 och 1.

Därefter startades spektralprogrammet och kurvan visades på bildskärmen. Ett försök till optimering gjordes på det sätt som beskrivs nedan, varefter ytterligare några körningar gjordes dels för att närmare undersöka de resultat man kommit fram till, dels för att jämföra resultaten med de värden testgeneratoren producerat.

När spektralprogrammet startats, skrevs kommandot $FX=5$, vilket innebär, att man antar, att kurvan består av Gausskurvor och en rät linje. $FX=4$ innebär, att man antar, att kurvan endast består av Gausskurvor, och är alltså ett specialfall av det förra (fallet den rätta linjen är $f(t)=0$ för alla t). Den rätta linjen (då $FX=5$) beskrivs vid bildskärmen som linjen genom (α_1, β_1) och (α_2, β_2) . För att starta med linjen $f(t)=0$ för alla t , sattes $\alpha_2=5$, medan övriga parametrar fick behålla sina ursprungsvärden 0. Efter dessa operationer fanns följande bild på skärmen:

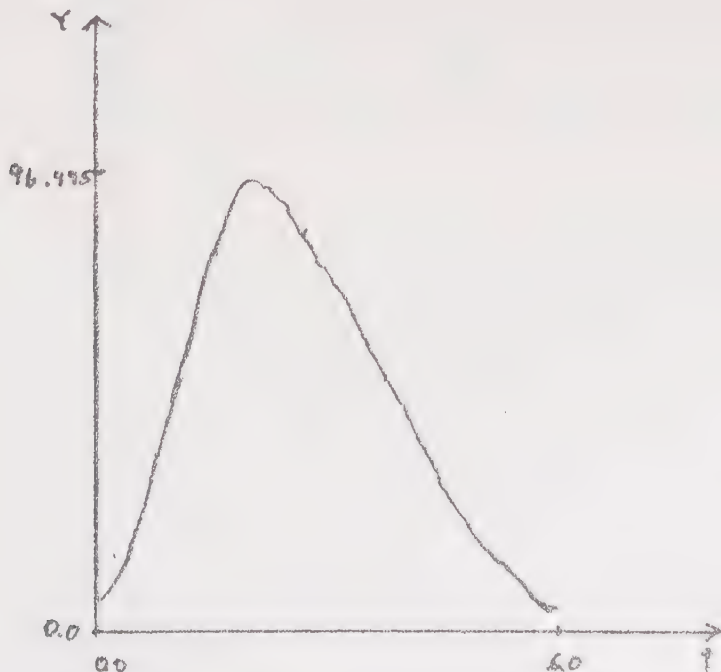


fig. 20

Det första kommandot till huvudmaskinen sändes nu. Det var
SKRIV=11112

Kommandona $FX=5$ och $\alpha_2=5$ åstadkom endast förändringar i bildskärmenheten. Som framgår av appendix 2 innebär kommandot SKRIV=11112 att man ville ha en ganska omfattande utskrift på radskrivaren efter körningen, vilket var önskvärt i detta fall, då man ville kunna följa beräkningsgången noggrant efter körningen. Därefter sändes kommandot

NYKURVA

varvid värdena på FX och α -, β - och γ -variablerna överfördes till huvudmaskinen och kurvan $f(t)=0$ för alla t sändes tillbaka till bildskärmen. Därefter gjordes ett försök att låta programmet gissa en topp genom kommandot GISSA

Vid detta tillfälle användes de tidigare översända värdena på FX och α -, β - och γ -variablerna. Resultatet blev:

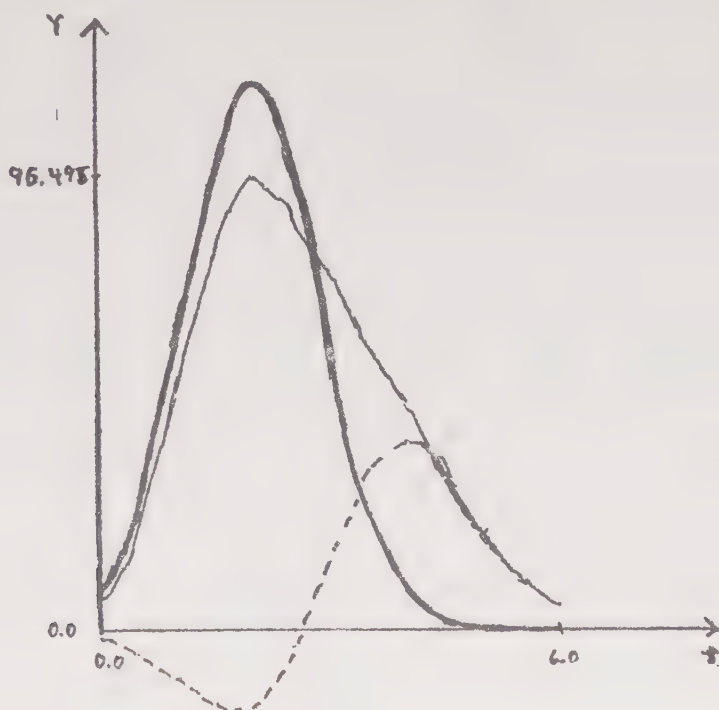


fig. 21

Den tjocka linjen är den gissade kurvan, den streckade differenskurvan.

De värden, som gissats var, $\alpha=112.25$, $\beta=1.9926$ och $\gamma=2.0562$. Som synes skiljer kurvorna sig ganska mycket från varandra. Emellertid kunde man senare konstatera, att anpassningen gjort efter den punkt där derivatan var störst, närmare bestämt kring punkten $t=1.3$. Den kurva, som man försökt gissa, hade varit den som producerats som $\alpha=67.790$, $\beta=1.8120$, $\gamma=1.8862$. Det var alltså framför allt α -värdet som var felaktigt. Den andra Gauss-kurvan, vars värden var $\alpha=54.746$, $\beta=3.3192$, $\gamma=2.7909$, påverkade resultatet avsevärt. Av en tillfällighet var det gissade värdet ungefär summan av Gauss-kurvornas α -värden.

Ött försök att gissa den andra kurvan gjordes trots resultatet ovan. Denna gång erhöles värden $\lambda=47.216$, $\mu=3.7571$, $\sigma=2.4264$. Kurvorna blev:

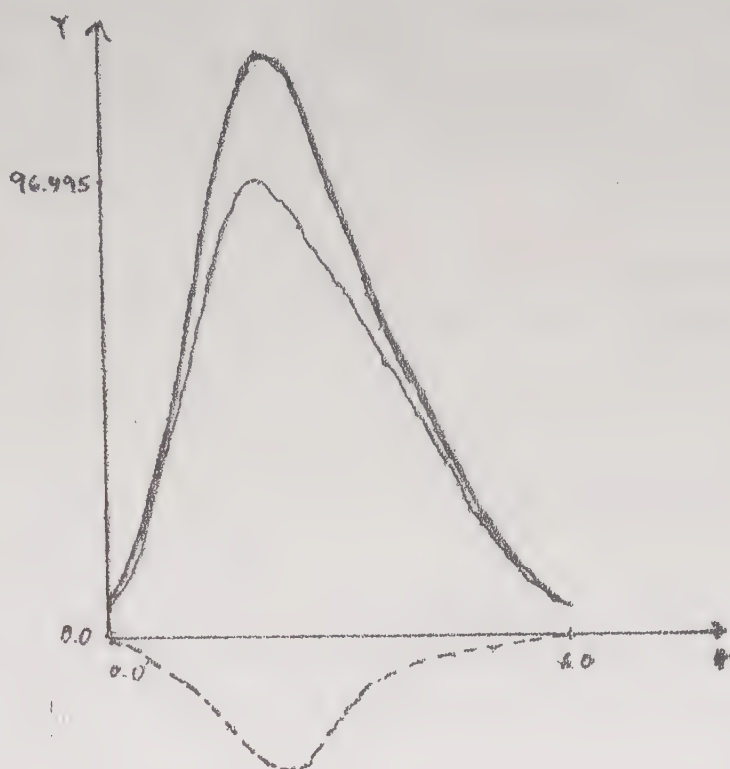


fig. 22

Nu försökte man göra en anpassning med Simplex-metoden. Med Simplex-metoden menas i denna rapport den metod, som beskrivits i Nelder-Mead (1965). Resultat:

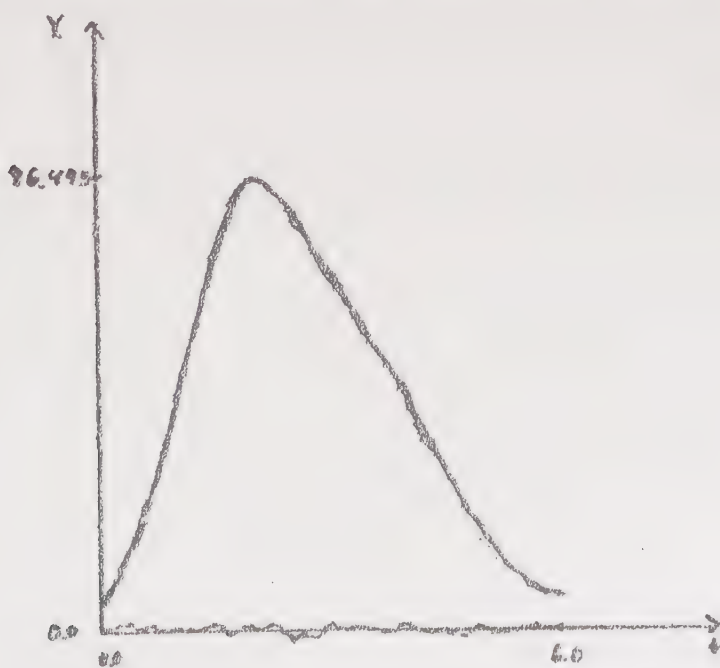


fig. 23

Det resultat, som skrevs på bildskärmen var: $\alpha_1=0$, $\beta_1=-1.8856$, $\alpha_2=5$, $\beta_2=3.2179$, $\alpha_3=92.567$, $\beta_3=2.0288$, $\gamma_3=2.1641$, $\alpha_4=37.431$, $\beta_4=3.8979$ och $\gamma_4=1.9560$. De första talparen betecknar, att linjen genom $(0, -1.8856)$ och $(5, 3.2179)$ adderats till de två Gauss-kurvorna $92.567 \cdot e^{-0(2.0288-t)^2/2.1641^2}$ och $37.431 \cdot e^{-c(3.8979-t)^2/1.9560^2}$, där $c=4 \times \ln(2)$. Felet, beräknat som $(\sum_{i=1}^n (y_1(i)-y_2(i))^2)/n$, minskade visserligen från 211.1 till 0.7551 (felet=0.3109 för den konstruerade kurvan), men Simplex-metoden gav "fel" minimum. (y_1 är originalkurvan, y_2 approximerande kurvan.)

Att det var fel kurva, dvs inte den kurva, som genererats, kunde man se på, att vissa parametervärden låg utanför det "riktiga" intervallet. Den rätta linjen skulle för $t=0$ och $t=5$ ligga mellan $y=-1$ och $y=1$, och α -värdena för Gauss-kurvorna skulle vara mellan 50 och 100. Om man inte vetat detta, hade det legat nära till hands att tro, att man funnit de sökta värdena. Exemplet kan dock tjäna som illustration på vilka svårigheter som kan uppstå vid anpassningen.

För att försöka få ett värde närmare det sökta sattes $\beta_1=0$ och $\beta_2=0$ och kommandot NYKURVA gjordes för att värdena skulle överföras till huvudmaskinen. Därefter anropades rutinen Stepit genom kommandot STEPIT vilket dock endast gav ett resultat i närheten av det Simplex-metoden givit ($\alpha_1=0$, $\beta_1=-3.0826$, $\alpha_2=5$, $\beta_2=1.0814$, $\alpha_3=91.892$, $\beta_3=1.9962$, $\gamma_3=2.1681$, $\alpha_4=39.831$, $\beta_4=3.8710$, $\gamma_4=2.1306$). Felet (som ovan) blev nu 0.65597, dvs kurvorna låg för farande nära varandra. Nu anropades Gauss-approximations-metoden genom kommandot GAUSSA

och resultatet blev $\alpha_1=0$, $\beta_1=0.2728$, $\alpha_2=5$, $\beta_2=0.5464$, $\alpha_3=68.851$, $\beta_3=1.8095$, $\gamma_3=1.8853$, $\alpha_4=54.149$, $\beta_4=3.3329$, $\gamma_4=2.7361$, vilket ligger innanför de på förhand uppställda gränserna. Kurvorna ligger mycket nära varandra (felet beräknat enligt ovan = 0.28799, dvs felet är mindre än det som konstruerats). Det absoluta felet funktionsvärdena $|y_1(i)-y_2(i)|$ var överallt mindre än 1. Beträffande parametervärdena var det största absoluta och relativa felet jämfört med den konstruerade kurva i α -värdena. α_3 var i den konstruerade kurvan ju 67.790 och $\alpha_4=54.746$. Vidare var $\beta_1=0$ och $\beta_2=0$.

Ytterligare ett försök att förbättra värdena gjordes genom att Gauss-approximationsmetoden närmast parameter ACCU minskades från 0.01 till 0.0001, varefter GAUSSA på nytt skrevs. Gaussa meddelade bara, att avrundningsfelet var större än förändringarna i parametervärdena och gav inte ett förbättrat resultat.

Slutresultatet säg ut på följande sätt:

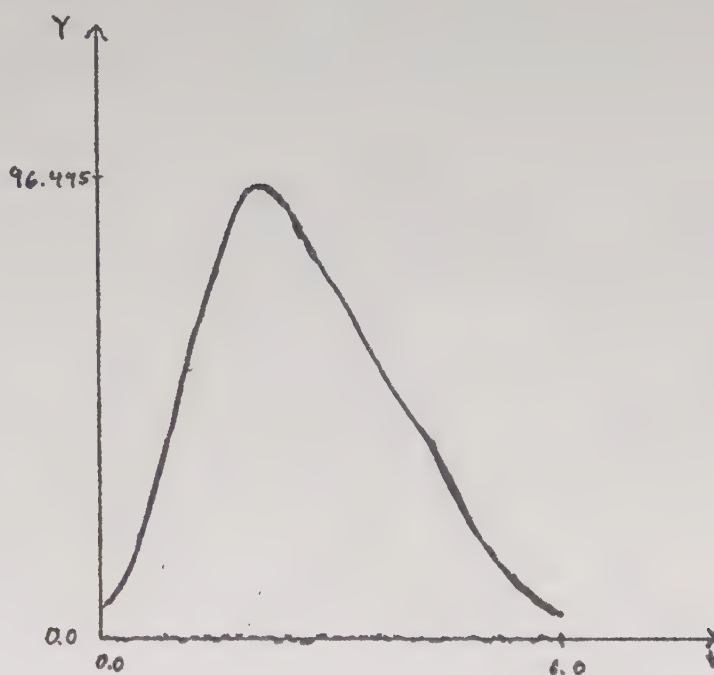


fig. 24

Efter detta gjordes några körningar för att ytterligare undersöka mellanresultat och slutresultat.

Efter att ha gissat toppar två gånger (fig. 22) kom man med Simplex-metoden till en annan punkt än den sökta. I denna förra punkten gjordes först ett Gauss-test, vilket lyckades, varpå rutinen Gaussa anropades. Gauss-approximationen fann direkt en lösning i närheten av slutresultatet av körningen, nämligen $\alpha_1=0$, $\beta_1=0.2719$, $\alpha_2=5$, $\beta_2=0.5469$, $\alpha_3=68.861$, $\beta_3=1.8095$, $\gamma_3=1.8854$, $\alpha_4=54.145$, $\beta_4=3.3331$ och $\gamma_4=2.7359$. Man gjorde också om anpassningen med Simplex-metoden med krav på större noggrannhet, nämligen genom att sätta parametern EPS till 0.001 (vilket innebär, att skillnaden mellan största och minsta funktionsvärdet till slut ska vara mindre än 0.001) och KMAX till 500 (högst 500 iterationer), men Simplex stannade fortfarande i närheten av den punkt den stannat på tidigare ($\alpha_1=0$, $\beta_1=-3.3667$, $\alpha_2=5$, $\beta_2=2.4736$, $\alpha_3=93.217$, $\beta_3=2.0105$, $\gamma_3=2.1765$, $\alpha_4=38.338$, $\beta_4=3.8939$ och $\gamma_4=2.0075$ och felet i denna punkt var 0.6533). När däremot Simplex anropades med samma startvärde, men med FX=4 (dvs endast två toppar utan möjlighet att lägga till en rät linje) och EPS=0.001 och KMAX=500, fann man ett bättre slutvärde. Det var $\alpha_3=61.262$, $\beta_3=1.7880$, $\gamma_3=1.8376$, $\alpha_4=54.149$, $\beta_4=3.3329$, $\gamma_4=2.9212$, övriga värden = 0 och felet = 0.2991. Jämfört med de värden testgeneratören använde har felet blivit något mindre (0.2991 mot 0.3109) och parametervärdena ligger i närheten av de önskade, möjligen med undantag för α_3 (61.262 i stället för 67.790).

Även på det värde Simplex-metoden givit (fig. 23) prövades Gauss-test. Testet lyckades, vilket tyder på, att värdet var i närheten av ett lokalt minimum. En efterföljande Gauss-approximation gav ett värde i närheten av det slutliga resultatet (största relativa skillnaden var ungefär 0.001).

På det slutliga värdet gjordes dels Gauss-test, vilket lyckades (tyder på att man är i närheten av minimum), dels anrop av såväl Simplex-rutinen som Stepit-rutinen, vilka endast förändrade resultaten i femte siffran i några enstaka fall.

Slutligen användes som startapproximation de parametervärden på α -, β - och γ -variablerna, som testgeneratoren använt. Körningarna gjordes dels med FX=5 (dvs med rät linje), dels med FX=4 (utan möjlighet att lägga till en rät linje), och i båda fallen anropades efter varandra rutinerna Simplex, Gauss, Stepit och Simplex igen.

Då FX var lika med 5, ändrade Simplex-rutinen värdena till $\alpha_1=0$, $\beta_1=.0024$ (tidigare 0), $\alpha_2=5$, $\beta_2=0.1897$ (tidigare 0), $\alpha_3=67.809$ (67.790), $\beta_3=1.8061$ (1.8120), $\gamma_3=1.8862$ (1.8862), $\alpha_4=54.703$ (54.746), $\beta_4=3.3151$ (3.3192) och $\gamma_4=2.7798$ (2.7909). Felet minskade från 0.3109 till 0.2905. Emellertid avbröts rutinen då det maximala antalet iterationer (=100) uppnåts. Med Gauss-approximation erhöles $\alpha_1=0$, $\beta_1=0.2725$, $\alpha_2=5$, $\beta_2=0.5451$, $\alpha_3=68.843$, $\beta_3=1.8094$, $\gamma_3=1.8853$, $\alpha_4=54.153$, $\beta_4=3.3328$, $\gamma_4=2.7363$, vilket är mycket nära det resultat som erhöles vid första körningen. Felet var 0.28798. Stepit förändrade inte detta resultat, Simplex ändrade γ -värdena till 1.8852 resp 2.7364.

Då FX var lika med 4, ändrade Simplex-rutinen värdena till $\alpha_3=67.786$, $\beta_3=1.8075$, $\gamma_3=1.8878$, $\alpha_4=54.747$, $\beta_4=3.3192$ och $\gamma_4=2.7892$, medan alla andra värden vid körningen är lika med 0. Felet minskades till 0.2934. Gauss-approximationen ändrade värdena till $\alpha_3=65.706$, $\beta_3=1.7990$, $\gamma_3=1.8709$, $\alpha_4=55.764$, $\beta_4=3.2782$, $\gamma_4=2.8301$ och felet till 0.2914, innan approximationen avbröts på grund av att avrundningsfelet var större än förändringarna i parametervärdena. Stepit förändrade inte detta resultat, Simplex ändrade γ -värdena till 1.8708 resp. 2.8302.

De av testgeneratoren framtagna Gauss-kurvorna hade följande utseende:

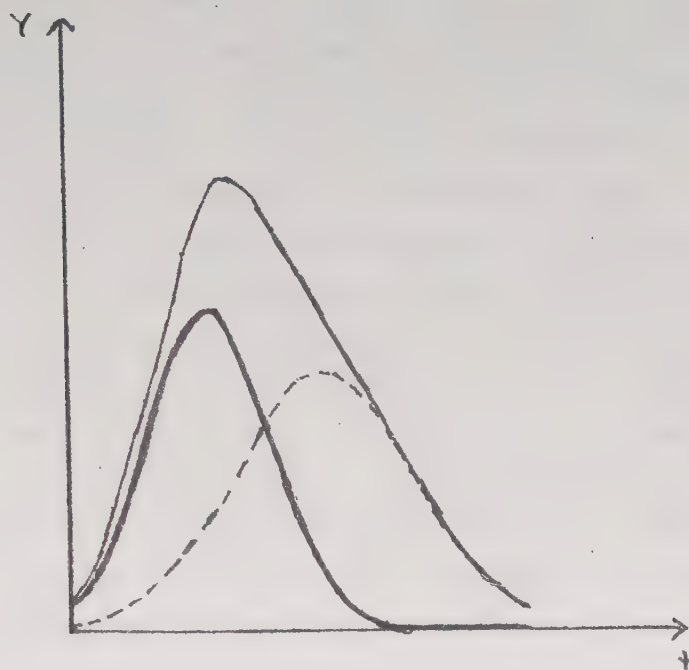


fig. 25

Den tjocka linjen är Gauss-kurvan med $\alpha=67.790$, $\beta=1.8120$ och $\gamma=1.8862$, den streckade linjen är Gauss-kurvan med $\alpha=54.746$, $\beta=3.3192$ och $\gamma=2.7909$ och den tunna linjen deras summa, dvs den av testgeneratoren producerade kurvan, men utan störningar.

Sammanfattning

I denna rapport beskrivs ett program för spektralanalys med hjälp av en grafisk bildskärm. Mätvärden, som erhålles vid spektralanalys, läses in av programmet, och motsvarande kurva ritas upp på bildskärmen. Kurvan betraktas som en summa av Gauss-kurvor eller Lorentz-kurvor och eventuellt en eller flera räta linjer, och man är intresserad av att bestämma vissa parametrar i dessa kurvor. Genom att skriva kommandon vid bildskärmenstangentbord kan man begära olika kurvor uppritade, och man kan jämföra dessa kurvor med kurvan som visar de ursprungliga mätvärdena. Om kurvorna ligger tillräckligt nära varandra, kan man med hjälp av olika optimeringsmetoder försöka förbättra överensstämmelsen mellan kurvorna. Man har för närvarande tre optimeringsmetoder att välja på. Vidare finns i programmet möjligheter att dra bort kurvor, när man fått tillräckligt god anpassning, och fortsätta med resten av kurvorna, hämta tillbaka resultat av tidigare gjorda anpassningar och eventuellt återgå till dessa vid de fortsatta anpassningarna, begära utskrift av kurvor och resultat på radskrivare, varvid omfattningen av utskrifterna bestäms genom kommandon via tangentbordet, m.m.

En fördel med detta program jämfört med andra program för datamaskinen på Lunds Datacentral är, att resultaten presenteras snabbare. En annan fördel är, att man inte går igenom en på förhand bestämd rutin eller ett antal olika rutiner i en bestämd ordning eller ett antal olika rutiner i en bestämd ordning eller en ordning bestämd av på förhand uppställda villkor, utan i stället bestämmer vilken rutin man ska ha med ledning av de mellanresultat man får. Det kan vara lättare för användaren vid bildskärmen än för maskinen att avgöra om kurvanpassningen är bra eller inte, eller om man t.ex. ska försöka lägga till eller dra bort en kurva.

För att lättare kunna testa programmet och underlätta analysen av resultaten har ett program skrivits, som genererar testdata. Redan efter ett litet antal försök fann man emellertid, att det är svårt att bestämma parametervärdena, beroende på, att felfunktionen man minimerar kan ha flera lokala minimipunkter och stationära punkter. Ett exempel på en sådan körning ges i rapporten.

Om ett lokalt minimum påträffas tycks samtliga optimeringsrutiner i programmet (Gauss-Newtons metod, Nelder-Meads simplexmetod och Chandlers subrutin Stepit) ha svårt att komma förbi minimipunkten och finna ett ännu mindre funktionsvärde. Ätminstone de två senare rutinerna tycks ha svårt att komma förbi stationära punkter. I en stationär punkt kan felfunktionen ha minimum då endast en parameter i taget ändras eller då vissa kombinationer av parametrar ändras. I de två senare metoderna börjar man med att bara ändra en parameter i taget och använder sig inte av derivator då en sökriktning ska bestämmas.

Ett enkelt exempel på en kurva i två dimensioner, som har en stationär punkt, är $f(x_1, x_2) = 2(x_1 - x_2)^2 - (x_1 + x_2)^2$ för $x_1 = 0, x_2 = 0$. Om x_1 hela tiden är lika med 0 fås $f(x_1, x_2) = -x_2^2$ med minimum då $x_2 = 0$, om x_2 hela tiden är lika med 0 fås $f(x_1, x_2) = x_1^2$ med minimum då $x_1 = 0$, men längs linjen $x_1 = x_2$, som går genom punkten $(0, 0)$, fås $f(x_1, x_2) = -(2x_1)^2$ med maximum då $x_1 = x_2 = 0$.

För närvarande planeras en undersökning av hur man ska göra för att komma förbi problemet med singulära punkter. I Gauss-Newton's metod övergår man i denna situation till att följa gradientriktningen. En annan metod, som ska undersökas, består i att man beräknar egenvärden och egenvektorer för en matris och minimerar längs en egenvektor med negativt egenvärde. Matrisen i fråga är $A^T A + \sum f_j G_j$, där f är funktionen som ska minimeras, A motsvarar förstaderivatorna och G andraderivatorna. Matrisen ingår i tredje termen i serieutvecklingen av $f^T(x+h)f(x+h) = f^T(x)f(x) + 2 f^T(x) A h + h^T (A^T A + \sum f_j G_j) h + \dots$, där $f(x+h) = f(x) + Ah +$

$$\frac{1}{2} \begin{pmatrix} h^T G_1 h \\ \vdots \\ h^T G_n h \end{pmatrix} + \dots, h \text{ söks.}$$

Om man fått ett lokalt minimum är det svårt att veta om minimum är globalt, dvs det sökta felet verkligen är så litet som möjligt. Ett problem är, att man knappast kan göra sig en föreställning om en funktion av mer än 3 variabler, och att antalet parameterkombinationer växer mycket snabbt då antalet parametrar ökas.

Vidare får man försöka bestämma när olika optimeringsmetoder bör användas och vilka kombinationer av metoder som fungerar bra i olika fall.

Appendix 1Lista över kommandon för spektralprogrammet

Kommandon skrivs på bildskärmsenhetens konsol ungefär som på en vanlig skrivmaskin. Varje kommando avslutas genom att man trycker på RETURN-tangenten. Hur kommandon rättas är beskrivet i appendix 3. Många kommandon kan skrivas i en kortare form. T.ex. testar programmet vid kommandot NY KURVA endast på de tre första bokstäverna, varför det räcker med att skriva NYK. I listan nedan stryks de tecken man måste ha med under. Vidare får blanktecken förekomma var som helst. T.ex. är kommandot $\alpha 1=1234$ likvärdigt med $\alpha 1 = 1\ 234$.

Följande typer av kommandon finns (ordnade i bokstavsordning fr.o.m. nr 3):

1. variabel=tal

där variabel kan vara enkel (endast namnet på variabeln skrivs)
 eller indicerad (skrivs variabelnamn (index) eller
 variabelnamn[index] eller i vissa fall variabelnamn index,
 där index ersätts med den sökta variabelns index)

= kan även skrivas := eller :

tal skrivs som tal i Algol eller Fortran (heltal eller reella tal)

ex:

FX=3

$\alpha 1=12$

Y(3):=-.5

$\gamma[5]=3E-2$ (Symbolen E (och &) betecknar "gänger tio upphöjt till")

$\gamma 5=0.03$ (likvärdigt med föregående kommando)

En lista över de variabler, som kan ändras, finns i appendix 2.

Om en variabel finns både i bildskärmsenheten och huvudmaskinen ändras endast variabeln i bildskärmsenheten, såvida man inte låter kommandot följas av ett nytt kommando EXTERN. (Undantag: FX, α -, β - och γ -värden överförs med kommandot NY KURVA (se nedan).
 Beträffande variabeln TEST, se detta kommando.)

variabel

där variabel kan vara enkel eller indicerad (jfr kommandot ovan). Kommandot innebär en förfrågan om en variabels värde. Variabelns värde skrivs ut på raden under tablån med α -, β - och γ -värden på bildskärmen. Om variabeln finns både i bildskärmsenheten och huvudmaskinen, och huvudmaskinens variabelvärde önskas, måste kommandot följas av ett nytt kommandot, EXTERN. En lista över de variabler man kan fråga efter finns i appendix 2.

Ex:

FX

MSLEFT

Y3(3)

Anm.: Om variabeln SPTEXT har värdet 0 tas kommandot bort när det har utförts, om SPTEXT har värdet 1 tas det inte bort. För att ta bort kommandot, tryck på en DELETE-tangent (se appendix 3).

Man kan även skriva ett kommando α , β eller γ följt av ett index. Efter detta kommando visas värdena på α_k , β_k och γ_k från och med det index, som står efter α , β eller γ och framåt, dock högst 5 indexvärden framåt. Högsta tillåtna index bestäms av en parameter EXPNR i programmet (denna variabel kan dock inte ändras från bildskärmen).

Ex:

α 1

α (5)

β [4]

Beträffande variabeln TEST, se detta kommando.

5. BILD bildnummer version

Kommandot innebär, att en bild, som sparats i bildskärmsenheten åter skall visas på skärmen.

Bildnummer skall vara numret på den bild, som skall visas (detta nummer står i övre vänstra hörnet på bilden). Man kan skriva bildnumret -1 för att få närmast föregående bild, -2 för att få den därpå föregående bilden etc., +1 för att få närmast efterföljande bild etc. Bildnumret ökas med ett varje gång en ny kurva sänds från huvudmaskinen. Bildnumret får omges av parenteser.

En bild kan sparas i flera versioner. Två versioner av samma bild skiljer sig från varandra genom att någon eller några α -, β - eller γ -variabler eller FX har tilldelats något nytt värde, medan kurvorna är oförändrade. Om en kurva sänds från huvudmaskinen kallas motsvarande bild version A. Om man skriver kommandot SPARA, ändras versionen från A till B, från B till C o.s.v. På samma sätt ändras versionen t.ex. efter kommandona NY KURVA, GAUSS och SIMPLEX. Om FX eller något α -, β - eller γ -värde ändrats, sparas bilden. Man kan utelämnas versionsbeteckningen och får då senaste sparade versionen av bilden.

4. DIFF

innebär, att differenskurvan (skillnaden mellan kurvorna på bildskärmen) används vid den fortsatta beräkningen. Genom detta kommando kan man t.ex. dra bort en bakgrund eller toppar, som man anpassat med tillräcklig noggrannhet.

Även om man ändrat antalet punkter man vill betrakta genom att ändra på variablerna MININD eller MAXIND, verkar operationen på så många punkter som man läst in från början. Den kurva, som senast beräknats, eventuellt på ett delintervall, beräknas för hela det ursprungliga intervallet, och differenskurvan för hela det ursprungliga intervallet beräknas och presenteras på bildskärmen. Dessutom ändras variablerna MININD, MAXIND, TSTART och TDIFF till sina ursprungliga värden.

5. DUMP

Kommandot gör, att den kurva man anpassar och den anpassande kurvan ritas upp på radskrivare, motsvarande funktionsvärden skrivs ut och aktuella α -, β - och γ -värden skrivs ut på radskrivare.

6. EJFIXERADE

Kommandot kan följas av en uppräknings av de variabler eller grupper av variabler, som i fortsättningen inte ska fixerade, dvs i fortsättningen ska kunna ändras om någon approximationsmetod anropas. Beträffande de variabelformer, som kan stå efter EJFIX, se kommandot FIX nedan. Om kommandot inte följs av något, är kommandot en förfrågan om vilka variabler, som inte är fixerade. I tabellen med α -, β - och γ -variabler visas talet 0 om variabeln inte är fixerad.

7. EXTERN

Kommandot används om man vill fråga efter eller ändra en variabel, som finns i huvudmaskinen, dvs är extern variabel ur bildskärmsenhetens synvinkel. Kommandot kan bara följa på ett kommando av typ 1 (variabel=tal) eller typ 2 (variabel). Om det gamla kommandot finns kvar på skärmen tages detta bort med en DELETE-tangent innan EXTERN-kommandot skrivs. En noggrannare beskrivning av EXTERN-kommandot har givits ovan (kommando av typ 1 och 2).

8. FEL

Kommandot medför, att huvudmaskinen gör en feluppskattning av α -, β - och γ -variablerna och sänder resultatet till bildskärmen. Där presenteras felen i tabellen med α -, β - och γ -värden. Om man åter vill ha de riktiga α -, β - och γ -värdena på skärmen, skriv t.ex. α följt av det index tabellen ska börja med (jämför kommando 2 ovan). Kommandot får bara skrivas omedelbart efter kommandot STEPIT eller efter kommandot STEPIT följt av kommandon som inte förändrar α -, β - eller γ -variablerna i 1108-an.

9. FIXERADE

Kommandot kan följas av en uppräknings av de variabler, som i fortsättningen ska vara fixerade, dvs i fortsättningen inte ska kunna ändras om någon approximationsmetod anropas. Variablerna bör åtskiljas med kommatecken. I stället för en variabel kan man skriva α , β , γ , δ eller ε , vilket gör, att samtliga α -, β -, γ -, δ - resp. ε -variabler i fortsättningen är fixerade. Man kan också byta ut en variabel mot ett tal. I så fall är alla variabler med motsvarande index fixerade i fortsättningen.

Ex:

FIX $\beta_3, 1, 2, \alpha, \gamma, \beta_5$

innebär, att samtliga α - och γ -variabler samt β_3 , β_5 , β_1 , γ_1 , ε_1 , β_2 , δ_2 och ε_2 i fortsättningen är fixerade.

Om kommandot inte följs av något, är kommandot en förfrågan om vilka variabler, som är fixerade. I tabellen med α -, β - och γ -variabler visas talet 1 om variabeln är fixerad. Om man åter vill ha tabellen med de verkliga α -, β - och γ -värdena skriver man t.ex. α följt av det index tabellen ska börja med (jämför kommando 2 ovan).

10. FIXERA EJ

Kommandot är likvärdigt med kommandot EJFIX och har samma **variabler** som detta kommando.

11. GAMMAL

Kommandot medför, att den kurva man anpassar, ersätts med den kurva man läst in från kortläsare (motsvarande). Den kurva man försöker anpassa ändras om man drar bort bakgrund eller några toppar med kommandot DIFF. Kommandot GAMMAL gör det möjligt att börja om från början igen om man gjort fel eller vill försöka approximera på något annat sätt. Kommandot återställer även värdena på MININD, MAXIND, TSTART och TDIFF, om de ändrats.

12. GAUSS

Kommandot ger anpassning med Gauss-approximation. Vid funktionsberäkningar används huvudmaskinens värden på FX och α -, β - och γ -variablerna, och den framräknade kurvan och de nya värdena på α -, β - och γ -variablerna sänds tillbaka och presenteras vid bildskärmen.

13. GAUSSTEST

eller

GAUSSTEST=heltal mellan 0 och 4

Kommandot likvärdigt med motsvarande kommando TEST (se nedan).

GISSA

Kommandot medför, att programmet i huvudmaskinen försöker gissa parametrarna α , β och γ till en Gauss-kurva på differenskurvan, lägger den gissade toppen till de tidigare Gauss-kurvorna, beräknar, sänder till bildskärmen och ritar upp den nya kurvan och de nya parametervärdena. Vid varje gissning ökas alltså antalet toppar med ett.

Programmet utnyttjar i detta fall, att om man har funktionen

$$f(t) = \alpha e^{-c(\beta-t)^2/\gamma^2}$$

så är $f'(t)/f(t)$ en rät linje, ur vars parametrar (k och l i $y=kt+l$) β och γ bestäms. Ur funktionen ovan bestäms därefter α .

Eftersom den kurva man undersöker ofta inte är så enkel som $f(t)$ ovan, försöker programmet endast göra en anpassning med minsta kvadratmetoden kring tre punkter i det intervall man betraktar, nämligen där kurvan har globalt maximum, där dess derivata har globalt maximum och där dess derivata har globalt minimum. Ur den, av dessa anpassningar, som lyckas bäst, bestäms värden på α , β och γ .

Om man i stället har funktionen

$$f(t) = \frac{\alpha}{1 + \gamma(\beta-t)^2 + \delta(\beta-t)^4 + \varepsilon(\beta-t)^6}$$

bestäms α och β på samma sätt som ovan, men γ beräknas som $1/\gamma^2$, δ som $1/(2 \cdot \gamma^4)$ och ε som $1/(6 \cdot \gamma^6)$ med γ som ovan, om c är lika med 1.

Dessa koefficienter har valts så att de motsvarar de första termerna i Taylor-utvecklingen: $f(t) = \alpha e^{-(\beta-t)^2/\gamma^2} = \alpha / [e^{(\beta-t)^2/\gamma^2}] = \alpha / [1 + (\beta-t)^2/\gamma^2 + (\beta-t)^4/(2! \cdot \gamma^4) + (\beta-t)^6/(3! \cdot \gamma^6) + (\beta-t)^8/(4! \cdot \gamma^8) + \dots]$.

Vid beräkningen används de värden på FX och α -, β - och γ -variablerna, som finns i huvudmaskinen. Om man direkt vill låta programmet göra en gissning på den kurva man läst in (lämpligen efter att ha dragit bort bakgrundskurvan), kan man ge FX det värde man vill ha (välja den funktionsrepresentation man vill ha), sätta alla α -värden lika med 0 och skriva kommandot NY KURVA innan man skriver kommandot GISSA.

Om programmet skall gissa fleratoppar i intervallet, får man skriva kommandot GISSA flera gånger.

NYA α

där NYA kan ersättas med NY eller NYTT

α kan ersättas med β eller γ .

Kommandot likvärdigt med kommandot TEST

NY INMATNING

där NY kan ersättas med NYTT eller NYA

Kommandot läser in en ny kurva från kortläsare (motsvarande).

Genom detta kan man i samma körning anpassa flera kurvor (flera spektra).

Hur inmatning från kortläsare gar till beskrivs i appendix 4.

17. NY KURVA

där ordet NY kan ersättas med ordet NYTT eller NYA.

Vid detta kommando sänds de värden på FX och α -, β - och γ -variablerna, som finns i bildskärmsenheten, till huvudmaskinen, och motsvarande kurva beräknas, sänds tillbaka till bildskärmen och ritas upp på skärmen.

18. RADSKRIVARUTSKRIFT

Kommandot likvärdigt med kommandot DUMP (se ovan).

19. SIMPLEX

Kommandot ger anpassning med Simplex-approximation. Vid funktionsberäkningar används huvudmaskinens värden på FX och α -, β - och γ -variablerna. (Dessa variabler får nya värden t.ex. vid kommandona NY KURVA, GAUSS och SIMPLEX.) Den framräknade kurvan och de nya värdena på α -, β - och γ -variablerna sänds tillbaka och presenteras vid bildskärmen.

20. SPARA

Kommandot medför, att den bild, som finns på skärmen sparas i bildskärmsenheten med samma bildnummer och närmast efterföljande versionsnamn, förutsatt att FX eller någon α -, β - eller γ -variabel tilldelats något nytt värde sedan föregående bild sparats.

Endast ett begränsat antal bilder sparas. När det minnesområde i bildskärmsenheten, som reserverats för att spara bilder, har fyllts, ersätts efterhand de tidigast sparade bilderna med nya bilder.

Bilderna kan sedan återfås på skärmen med kommandot BILD (se detta kommando ovan).

21. STANNA

Detta kommando medför, att ett meddelande sänds till huvudmaskinen, varefter förbindelsen med huvudmaskinen bryts, denna del av programmet avslutas och resultaten som efterhand framkommit vid körningen skrivs ut på radskrivare. Om man efter detta vill använda programmet i bildskärmen för ytterligare en körning, skriver man kommandot NY KURVA och startar därefter på nytt programmet i huvudmaskinen.

22. STEPIT

Kommandot ger anpassning med rutinen Stepit. Vid funktionsberäkningar används huvudmaskinens värden på μ -, ρ - och γ -variablerna.

Den framräknade kurvan och de nya värdena på μ -, ρ - och γ -variablerna sänds tillbaka och presenteras vid bildskärmen.

23. STOPPA

Kommandot likvärdigt med STANNA

24. TEST

eller

TEST=heltal mellan 0 och 4.

Kommandot bestämmer, om ett test skall utföras för att avgöra valet mellan approximationsmetoderna, då kommandot GAUSS skrivs.

Om TEST får värdet 0 eller 1 utförs inte testet, och Gauss-approximation görs.

Om TEST får värdet 2, 3 eller 4 utförs testet. Om testet lyckas utförs Gauss-approximation. Om testet misslyckas, sänds ett meddelande om detta till bildskärmen. I detta fall görs ingen approximation om TEST har värdet 2, Simplex-approximation om TEST har värdet 3 och Gauss-approximation om TEST har värdet 4.

Kommandot TEST är likvärdigt med kommandot TEST=3.

25. UTJAMNA $v_1, v_2, \dots, v_k$

där v_i är positiva heltal

Kommandot medför, att ojämnheter på den kurva, som skall anpassas, jämnas ut genom att varje punkt y_p ersätts med ett viktat medelvärde

$$\frac{(y_p v_1 + (y_{p+1} + y_{p-1}) v_2 + \dots + (y_{p+k} + y_{p-k}) v_k)}{v_1 + 2v_2 + \dots + 2v_k}$$

(I ytterkanterna av intervallet modifieras formeln något.)

k (antalet vikter) antages ligga mellan 2 och 10.

Man kan också skriva kommandot utan vikter. I så fall används samma vikter som vid föregående utjämning (startvikter $v_1=2$, $v_2=1$, dvs varje punkt ersätts med $0.25y_{p-1} + 0.5y_p + 0.25y_{p+1}$).

26. VIDGA

Kommandot sätter MININD=1 och MAXIND=MAXN, dvs ett eventuellt begränsat intervall ersätts med det intervall man använde vid inläsningen av kurvan. Den nya kurvan sänds tillbaka och ritas upp på bildskärmen.

Även TSTART och TDIFF återfår sina utsprungliga värden.

27. (blank rad)

Kommandot tar bort eventuella felmeddelanden från bildskärmen.

Appendix 2

Lista över variabler, som kan avläsas eller ändras från bildskärmen

Man kan genom att skriva en variabels namn eller dess namn följt av likhetstecken och ett tal få en variabels värde utskrivet på bildskärmen resp. dess värde ändrat. Kommandona är beskrivna i appendix 1, kommando 1 och 2. Denna lista över variabler är ordnad i bokstavsordning, men om en variabel är beskriven i samband med ett annat variabelnamn ersätts beskrivningen av variabeln med en hänvisning till det andra namnet.

ACCU

ACCU anger noggrannheten i Gauss-approximationen. Man kan använda variabeln så att man börjar med ett stort värde på ACCU, dvs nöjer sig med liten noggrannhet, och ger ACCU ett litet värde när man är mycket nära en bra lösning, varefter man på nytt försöker göra en Gauss-approximation.

AL

AL, BE och GA är programmets motsvarigheter till konstanterna Alfa, Beta resp. Gamma i Simplex-approximationen, dvs AL är reflektionskonstanten (x_h ersätts med $x_r = (1 + AL)\bar{x} - AL x_h$) BE kontraktionskonstanten (x_h ersätts med $BE y + (1 - BE)\bar{x}$, där $y = x_h$ eller x_r) och GA expansionskonstanten (x_h ersätts med $GA x_r + (1 - GA)\bar{x}$).

ALPOUT

Variablerna ALPOUT, CHAOUT, IOOUT, KLAOUT, YOUT, YSPACE, IOC, IPRINT, NTRACE, FIXOUT, BACKUP och SKRIV bestämmer omfånget av radskrivarutskrifterna. Om ALPOUT har värdet 1 skrivs värdena på α , β och γ -variablerna ut när de ändras, om ALPOUT har värdet 0 skrivs de inte ut. Om CHAOUT har värdet 1 sker utskrift när variabler i huvudmaskinen efterfrågas eller ändras från bildskärmen, om CHAOUT har värdet 0 skrivs detta inte ut. Om KLAOUT har värdet 1 skrivs värdet av en variabel KLAR ut, om dess värde är skilt från 0 efter att Gauss-approximation gjorts, vilket innebär, att något fel uppstått vid approximationen. Om man begärt, att ett test skall utföras innan man gör Gauss-approximation, skrivs resultatet av detta ut om KLAOUT är 1 (se kommandot TEST i appendix 1). Om KLAOUT är 0 sker inte någon av dessa utskrifter. Om IOOUT har värdet 2 skrivs allt som sänds till och från bildskärmen ut som oktala heltal och som decimala heltal, med undantag för eventuella avslutande nollor vid sändningarna. Om IOOUT har värdet 1 skrivs de decimala heltal, som sänds till och från bildskärmen ut, med undantag för avslutande nollor i sändningarna.

Om IOOUT har värdet 0 skrivs sändningarna inte ut. Om IOOUT har värdet -1 skrivs endast första talet vid varje sändning ut. Detta tal är en kod, som beskriver typen av sändning. Om IOOUT har värdet 3 sker utskrift som IOOUT lika med 2, men dessutom sändningar, som endast bekräftar, att föregående sändning mottagits (t.ex. en sändning från bildskärmen mellan två på varandra följande sändningar från huvudmaskinen). Om YOUT har värdet 1 skrivs funktionsvärden på de kurvor, som sänds till bildskärmen, ut. Om YOUT är 2 eller 3 ritas kurvorna också upp på radskrivare, varvid man i det senare fallet alltid börjar på ny sida och undviker avbrott i figuren, vilka annars uppstår då en sida är slut. Om YOUT har värdet 0 sker inte någon sådan utskrift. Om YOUT har värdet -1 sker endast utskrift av vissa mellanresultat då en topp gissas. Om YOUT är lika med 2 eller 3 kan man med YSPACE närmare bestämma hur utskriften av funktionsvärdena ska se ut. Om YSPACE har värdet 0 skrivs varje värde ut med ett värde per rad. Om YSPACE är större än 0 lämnas YSPACE blanka rader mellan varje utskriftsrad. Exempelvis ger YSPACE = 1 utskrift på varannan rad. Om YSPACE är mindre än 0 hoppas efter varje utskrivet värde YSPACE värden över. Exempelvis ger YSPACE = -2 utskrift av vart tredje funktionsvärde. IOC bestämmer omfattningen av utskrifter då Gauss-approximation, Gauss-test eller sökning efter kommandot SÖK görs. Om IOC är 0 skrivs inga resultat ut. Om IOC är lika med 1 skrivs värdet av en variabel PHI ut i varje steg och om IOC är lika med 2 skrivs mer detaljerad utskrift ut i varje steg, om Gauss-approximation görs. Om IOC är större än 0 skrivs sökriktningar, steglängder och funktionsvärden ut, då sökning görs. Om IOC har värdet 3 skrivs funktionsvärden och värden av partiella derivator ut varje gång de beräknas. IPRINT styr utskriften vid anrop av Simplex-approximation. Om IPRINT är 0 skrivs ingenting ut. Om IPRINT är -1 skrivs funktionsvärden ut, om IPRINT är 2 skrivs start- och slutvärden ut, och om IPRINT är lika med 1 skrivs en detaljerad utskrift ut i varje steg. NTRACE styr utskriften vid anrop av rutinen Stepit och vid felberäkningen efter avrop av Stepit. Om NTRACE har värdet -1 sker inga utskrifter, om NTRACE har värdet 0 skrivs startvärdena och slutvärdena ut, och om NTRACE är lika med 1 skrivs detaljerad information i varje steg ut. Om FIXOUT har värdet 1 skrivs en tabell ut, som visar vilka variabler som är fixerade, varje gång variabler fixeras eller slutar vara fixerade, om FIXOUT har värdet 0 sker inte dessa utskrifter. α -, β -, γ -, δ - och ε -variablerna kan fixeras, vilket innebär, att de

inte ändras då någon approximationsmetod anropas. Om BACKUP har värdet 1, sker utskrift då en kurva läses in eller ändras genom att differenskurvan tas som ursprungskurva. Om BACKUP är lika med 0 sker inte denna utskrift. Med hjälp av SKRIV kan man avläsa eller ändra värden på flera av ovanstående variabler samtidigt. SKRIV är lika med $10000 \cdot \text{ALPOUT} + 1000 \cdot \text{CHAOUT} + 100 \cdot \text{IOOUT} + 10 \cdot \text{KLAOUT} + \text{YOUT}$. Om man t.ex. frågar efter värdet SKRIV och får svaret 11010, så innebär det, att ALPOUT är 1, CHAOUT är 1, IOOUT är 0, KLAOUT är 1 och YOUT är 0. Om man skriver SKRIV=1000 vid bildskärmen, ges ALPOUT, IOOUT, KLAOUT och YOUT värdet 0, CHAOUT värdet 1. Om man försöker ge en variabel ett otillåtet värde, ändras inte det gamla värdet, t.ex. SKRIV=99092 ger endast IOOUT värdet 0 och YOUT värdet 2. Kommandot SKRIV=0 ger dessutom IOC, IPRINT, FIXOUT och BACKUP värdet 0 och NTRACE värdet -1, (ingen utskrift alls i fortsättningen, om utskriften regleras av någon variabel), kommandot SKRIV=11111 ger dessutom IOC och IPRINT värdet 2, FIXOUT och BACKUP värdet 1 och NTRACE värdet 0 (ganska omfattande utskrifter i fortsättningen).

BACKUP

Se ALPOUT

BE

Se AL

C

Konstanten c, då FX har värdet 4 eller 5, dvs c i någon av funktionsformerna

$$y = \sum_k \alpha_k e^{-c(\beta_k - t)^2 / \gamma_k^2}$$

eller

$$y = \sum_{k=3}^n \alpha_k e^{-c(\beta_k - t)^2 / \gamma_k^2} + \text{den rätta linje som går genom } (\alpha_1, \beta_1) \text{ och } (\alpha_2, \beta_2)$$

Se även beskrivning av FX i detta appendix och programbeskrivningen ovan.

CHAOUT

Se ALPOUT

DNR

DNR och DY används då man ska gissa en ny kurva (se kommandot GISSA). Derivatn i y_k beräknas som $(y_{k+dy} - y_{k-dy}) / (2 \cdot dy \cdot \text{tdiff})$, vid beräkning av minsta-kvadrat-summan används $2 \cdot \text{dnr}$ värden.

DY

Se DNR

EPS

EPS anger noggrannheten vid Simplex-approximation. (Jämför med ACCU.)

EPS1

EPS1, EPS2, EPS3, EPS4 och EPS5 anger preliminära steglängder för α -, β -, γ -, δ - resp ε -värdena då Simplex-approximation ska utföras. Innan approximationsrutinen anropas görs ett försök att förbättra steglängderna.

EPS2

Se EPS1

EPS3

Se EPS1

EPS4

Se EPS1

EPS5

Se EPS1

FAST

Variabeln bestämmer vilken algoritm som ska användas vid funktionsberäkningarna. Om FAST har värdet 0 beräknas för varje punkt ($t=t_1, t_2, \dots, t_n$, med $t_1=t_{start}$ och $t_{k+1}=t_k+tdiff$ (konstant differens))

$$y(t) = \sum_{k=1}^r a_k e^{-(\beta_k - t)^2 / \gamma_k^2}$$

med algoritmen (uttryckt i Algol-liknande notation):

t:=tstart;

for i:=1 step 1 until n do

begin sum:=0;

for k:=1 step 1 until r do sum:=sum+a[k]*exp((($\beta[k]$ -t)/ $\gamma[k]$)²);

y[i]:=sum;

t:=t+tdiff

end

(om man antar, att r Gauss-kurvor skall beräknas och funktionsformen ser ut som ovan).

Med n punkter och r Gauss-kurvor anropas exponentialfunktionen n*r gånger då en kurva beräknas.

Om FAST har värdet 1 används en algoritm, som gör ett mindre antal exponentialberäkningar och som därför troligen är något snabbare, men som möjligen kan ge något mindre exakta funktionsberäkningar genom att avrundningsfel kan ackumuleras. Felet kan bara ha betydelse om man är mycket nära en bra lösning.

Denna algoritm utnyttjar, att om $f(t)$ är en linjär funktion, dvs $f(t)=k \cdot t+1$

så är $f(t+tdiff)=k \cdot (t+tdiff)+1=f(t)+k \cdot tdiff$,

där $k \cdot tdiff$ är en konstant, dvs ett funktionsvärde fås genom att man adderar en konstant till föregående funktionsvärde.

På liknande sätt visas lätt, att om

$$f(t)=a \cdot e^{b \cdot t}$$

så är $f(t+tdiff)=f(t) \cdot \text{konstant}$, dvs ett funktionsvärde fås genom att

man multiplicerar föregående med en konstant. I detta fall behöver man inte göra n exponentialberäkningar för att beräkna n funktionsvärden, utan endast en exponentialberäkning och $n+1$ multiplikationer (inklusive två multiplikationer för $t=tstart$).

Om man har funktionen

$$f(t)=a \cdot e^{b(c-t)^2}$$

så är $f(t+tdiff)=f(t) \cdot g(t) \cdot a_1$, där $g(t)=a_2 \cdot e^{b_1 \cdot t}$ (a, a_1, a_2, b, b_1 och c är konstanter). Genom att utnyttja detta behövs endast tre multiplikationer för att beräkna ett funktionsvärde ur det föregående.

Med samma funktion som tidigare blir algoritmen:

for $k:=1$ step 1 until r do

begin

$t1:=(tstart-x[k])/y[k]; t2:=tdiff/y[k];$

$f1:=x[k] \cdot \exp(-t1^2); e1:=\exp(-t2^2);$

$e2:=e1^2; \text{comment } e2:=\exp(-2 \cdot t2^2);$

$e3:=\exp(-2 \cdot t1 \cdot t2);$

for $i:=1$ step 1 until n do

begin $y[i] := \text{if } k=1 \text{ then } f1 \text{ else } y[i]+f1; f1:=f1 \cdot e3 \cdot e1; e3:=e3 \cdot e2$ end;

end

Med denna algoritm görs 3·r exponentialberäkningar.

FIXOUT

Se ALPOUT

FX

FX används för att bestämma vilken funktionsform, som skall användas vid funktionsberäkningar. Värdet på FX finns i bildskärmsenhetens program, och dess värde överförs till huvudmaskinen vid kommandot NY KURVA. De 8 funktionsformerna är beskrivna i ett tidigare avsnitt (formel 3.1-9, se ovan). Dessutom finns en möjlighet att ge FX värdet 7 och skriva kommandot NY KURVA. I detta fall överförs värdena på α -, β - och γ -variablerna utan att någon funktionsberäkning utförs, och huvudmaskinens värde på FX förblir oförändrat och detta värde sänds tillbaka till bildskärmen och återställer bildskärmens värde på FX. Detta kan användas om man t.ex. gjort en approximation med en

metod och vill pröva en annan metod med exakt samma värden på α -, β - och δ -variablerna. Man kan då begära tillbaka en tidigare bild vid bildskärmen, ge FX värdet 7, skriva NY KURVA och den andra approximationsmetodens namn. (Annars används vid Gauss-approximation och Simplex-approximation som startapproximation de värden på FX och α -, β - och δ -variablerna, som användes då senaste kurvan sändes till bildskärmen.)

Anpassning med någon approximationsmetod kan endast ske med någon av funktionsformerna 1 - 5 eller 8 - 9. Resultatet av en approximation innehåller lika många Gauss-kurvor, som den startapproximation man givit (dock minst 1 och högst så många som bestäms av parametrar i programmet, för närvarande 10). Om man vill göra anpassning till t.ex. två Gauss-kurvor och FX är mindre än 5, ska exakt två q -värden vara skilda från 0.

GA

Se AL

IOC

Se ALPOUT

IOOUT

Se ALPOUT

IPRINT

Se ALPOUT

KLAOUT

Se ALPOUT

KMAX

Maximala antalet iterationer, som utförs vid en Simplex-approximation.

LX

LX, LX1, LY, LY1, UX, UX1, UY, UY1, ZX och ZY är heltal, som anger koordinater för bildskärmen (i rutnätet med 1024 x 1024 punkter på skärmen) som används då kurvorna och koordinataxlarna skall ritas upp. Den kurva man skall anpassa skall befinna sig inom rektangeln (LX,LY), (LX,UY), (UX,UY), (UX,LY). Koordinataxlarnas skärningspunkt är (ZX,ZY). Vid punkten (LX1,ZY) står minsta x-värdet på kurvan (som reellt tal), vid punkterna (UX1,ZY), (ZX,LY1), (ZX,UY1) motsvarande största x-värde, minsta y-värde och största y-värde för den kurva man skall anpassa.

LX1

Se LX

LY

Se LX

LY1

Se LX

MATRIX

MATRIX används vid felberäkningen efter rutinen Stepit. Om MATRIX är mindre än 80 eller större än 130 görs ingen felberäkning. Annars används vid felberäkningen en steglängd $= DX \cdot 10^{(MATRIX-100)}$.

I detta fall bör MATRIX ha ett värde mellan 100 och 110.

MAXIND

När man läser in en kurva från kortläsare (motsvarande), numreras punkterna 1, 2, ... , MAXN (dvs MAXN får värdet av antalet inlästa punkter). Denna kurva kan sedan begränsas genom att man talar om numret på de punkter man vill börja resp. sluta med. Det förra numret betecknas med MININD, det senare med MAXIND. Antalet punkter på den kurva, som ritas upp på bildskärmen efter att MININD eller MAXIND ändrats, är N. Om MININD eller MAXIND ändras, sätt TSTART lika med $X(\text{MININD})$ och TDIFF lika med $X(\text{MININD}+1)-X(\text{MININD})$, och de nya kurvorna beräknas och ritas upp på bildskärmen. Vid kommandona VIDGA, DIFF och GAMMAL ändras även värdena på MININD och MAXIND till 1 resp. MAXN, och TSTART och TDIFF till $X(1)$ resp. $X(2)-X(1)$. Man kan i och för sig även ändra värdena på TSTART och TDIFF från bildskärmen, vilket innebär, att x-värdena ändras utan att kurvorna flyttas, vilket kan användas så att man inte behöver skriva så många siffror när man gissar värden på α -, β och γ -variablerna, men i så fall måste man ändra tillbaka några av dessa variabler om man ändrar MININD, eller MAXIND (eller gör något av kommandona VIDGA, DIFF eller GAMMAL).

MAXN

Se MAXIND

MAXSTE

MAXSTE och MINSTE används då rutinen Stepit anropas. Som ursprungliga steglängder för α_k , β_k och γ_k används MAXSTE $\cdot |\alpha_k|$, MAXSTE $\cdot |\beta_k|$ resp. MAXSTE $\cdot |\gamma_k|$, som minsta steglängder används MINSTE $\cdot |\alpha_k|$, MINSTE $\cdot |\beta_k|$ resp. MINSTE $\cdot |\gamma_k|$. Om någon steglängd skulle bli lika med 0, ersätts den med MAXSTE resp. MINSTE.

MAXT

Maximal tid i millisekunder, som en Gauss-approximation får ta, sätts till (antalet funktionsvärden $\cdot$ antalet parametrar som optimeras² $\cdot 2$), dock högst MAXT och minst MINT millisekunder. Normalt slut vid Gauss-approximation är, att noggrannheten ACCU uppnås (se ACCU ovan).

MININD

Se MAXIND

MINSTE

Se MAXSTE

MINT

Se MAXT

MSLEFT

När man kör programmet anges på första styrkortet i den kortbunt, som startar programmet i huvudmaskinen, hur lång tid beräkningarna i huvudmaskinen högst får ta sammanlagt. Genom att skriva MSLEFT får man veta, hur mycket av denna tid som återstår (tiden mäts i millisekunder).

N

Se MAXIND

NEMAX

Maximalt antal funktionsberäkningar i rutinen Stepit

NN

Likvärdigt med N (Se MAXIND).

NTRACE

Se ALPOUT

SEK

Om man inte gjort något anrop från bildskärmen till huvudmaskinen på 10 minuter, avbryts programmet i huvudmaskinen. (Kontrollen inlagd som säkerhetsåtgärd för att programmet i huvudmaskinen säkert skall avslutas även om man glömmer skriva kommandot SLUTA när man lämnar bildskärmen eller förbindelsen mellan maskinerna av annan anledning avbryts på onormalt sätt. Tidsgränsen kan ändras om man ändrar i programmet.) SEK anger tiden i sekunder från senaste anrop till huvudmaskinen.

SKRIV

Se ALPOUT

SPTEXT

Om man frågar efter en variabels värde, tages frågan bort, då man får svar, om SPTEXT är lika med 0. Om SPTEXT är lika med 1 står variabeln kvar på bildskärmen då frågan besvarats och kan tagas bort genom att man trycker på en DELETE-tangent.

TDIFF

Se MAXIND

TOTSEK

Ungefärligt värde på totala antalet sekunder programmet i bildskärmen varit igång (sedan programmet startades). (Om man ger TOTSEK värdet 0 när man börjar approximera en ny kurva, kan man istället mäta, hur lång tid man behöver för att göra anpassningen.)

TSTART

Se MAXIND

UX

Se LX

UX1

Se LX

UY

Se LX

UY1

Se LX

X

X är ett fält, som innehåller x-koordinaterna (x-värdena) på de kurvor, som finns på bildskärmen. Värdet i bildskärmen anger koordinaten i x-led i rutnätet på 1024 x 1024 punkter, värdet i huvudmaskinen anger funktionens x-värde som ett reellt tal. Om man kallar fältet X eller X1 får man i första hand värdet på bildskärmskoordinaten, i andra hand värdet i huvudmaskinen, medan namnet XEXT direkt undersöker huvudmaskinens värde (externa x-värdet, sett från bildskärmsenheten). Efter namnet skall det stå parentes och index (numret på den punkt man efterfrågar eller vill ändra). Den kurva, som skall anpassas, har namnen Y, Y1 (för bildskärmskoordinaterna) och YEXT (för huvudmaskinens y-värden) på motsvarande sätt. Den anpassande kurvan har namnet Y2, differenskurvan namnet Y3.

X1

Se X

XEXT

Se X

Y

Se X

Y1

Se X

Y2

Se X

Y3

Se X

YEXT

Se X

YOUT

Se ALPOUT

YSPACE

Se ALPOUT

ZX

Se LX

ZY

Se LX

Appendix 3

Rättning av kommandon

Kommandon ges från bildskärmsenheten genom att man skriver dem på ett tangentbord vid bildskärmen ungefär som på en vanlig skrivmaskin. De flesta tecken, som man skriver, lagras i en skrivarean och visas på bildskärmen. Innehållet i skrivarean tolkas som ett kommando när man trycker på RETURN-knappen. Om ett kommando accepteras, fylls skrivarean med blanktecken och man kan börja skriva ett nytt kommando. Om kommandot inte accepteras, förblir skrivarean oförändrad och ett felmeddelande tänds på raden ovanför skrivarean. Ett sådant felmeddelande tas bort när ett kommando accepteras. Speciellt accepteras en blank rad som kommando. Om man skriver fel kan felet rättas på olika sätt. Man kan ta bort hela raden med någon av tangenterna ERASE LINE eller DELETE LINE. Man kan förflytta sig till vänster och höger utan att ändra redan inslagna tecken (t.ex. för att ändra ett tidigare tecken) med hjälp av tangenterna ← resp. →. Tangenterna på högra tangentbordet ger ett avbrott varje gång man trycker på tangenterna, medan tangenterna i nedre vänstra hörnet ger 3-4 avbrott per sekund med jämna mellanrum. Vid förflyttning till vänster tas tecken bort från skärmen, vid förflyttning därefter till höger återkommer de.

Appendix 4

Beskrivning av några numeriska metoder, som används i programmet

GAUSS-APPROXIMATION

Programmet har skrivits av P. Å. Wedin och L. Å. Carlsson vid Lunds Datacentral, och finns i Lunds Datacentrals programbibliotek med namnet GAUSSA. Programmet är beskrivet i motsvarande programdokumentation. Metoden diskuteras i Ruhe: Föreläsningar om numerisk behandling av matriser, Ruhe och Wedin: Algorithms for separable nonlinear least squares problems, Wedin: The non-linear squares problems from numerical point of view och Wedin: On the Gauss-Newton method for the non-linear least squares problem.

Då metoden används, ska en lösning till ett icke-lineärt minsta kvadratproblem beräknas genom att succesivt bättre approximationer beräknas med Gauss metod. Man approximerar då med linjära minsta kvadratproblem och använder Armijos steglängdsprincip i varje steg.

Metoden kan också beskrivas på följande sätt:

Vi antar, att vi har mätvärden $y_1(1), y_1(2), \dots, y_1(n)$ för $t=t_1, t_2, \dots, t_n$ och approximerar $y_1(i)$ med $y_2(i) = \sum_{k=1}^r \alpha_k e^{-(\beta_k - t_1)^2 / \gamma_k^2}$.

Vi betraktar var och en av de n funktionerna $f_1 = y_1(i) - y_2(i)$ som funktion av $3 \cdot r$ (=m) variabler $\alpha_1, \beta_1, \gamma_1, \dots, \alpha_r, \beta_r$ och γ_r .

Vi ska minimera $\phi = \sum_{i=1}^n rw(i) \cdot f_1^2$, där rw är en viktsfunktion (för närvarande identiskt lika med 1).

Låt f beteckna vektorn med funktionerna $(f_1, f_2, \dots, f_n)^T$. För varje kombination (m -tupel) av värden $\alpha_1, \dots, \gamma_r$ finns n värden $f_1, \dots, f_n$, dvs f är en funktion från R^m till R^n (R^n är det n -dimensionella rummet). Att minimera ϕ innebär (då $rw \equiv 1$) detsamma som att hitta $\alpha_1, \dots, \gamma_r$ som ger en punkt så nära punkten $(f_1=0, f_2=0, \dots, f_n=0)$ som möjligt, dvs så nära origo i det n -dimensionella rummet som möjligt.

Låt x beteckna en vektor $(\alpha_1, \dots, \gamma_r)$ i det m -dimensionella rummet. Vi vill bestämma en vektor h i det m -dimensionella rummet, som beskriver förändringen i $\alpha_1, \dots, \gamma_r$, och som är sådan, att $f(x+h)$ är så liten som möjligt. Vidare är $f(x+h)$ approximativt lika med $f(x) + Ah$, där A är en $n \times m$ -matris med $a_{ik} = \frac{\partial f_i}{\partial x_k}$. Tangentplanet genom en punkt $x=x_0$ kan beskrivas som $v = f(x_0) + A(x_0)h$ för godtyckliga vektorer h i R^m . På detta tangentplan projiceras vektorn från $f(x_0)$ till origo i det n -dimensionella rummet, och man får då en vektor i det m -dimensionella rummet ($= Ah$ ovan). Om $m=2$ (vi betraktar 2 variabler, t.ex. α_1 och β_1)

och $n = 3$ (3 punkter f_1 , f_2 och f_3 motsvarande 3 olika t -värden t_1 , t_2 resp. t_3) kan det hela illustreras av följande figur:

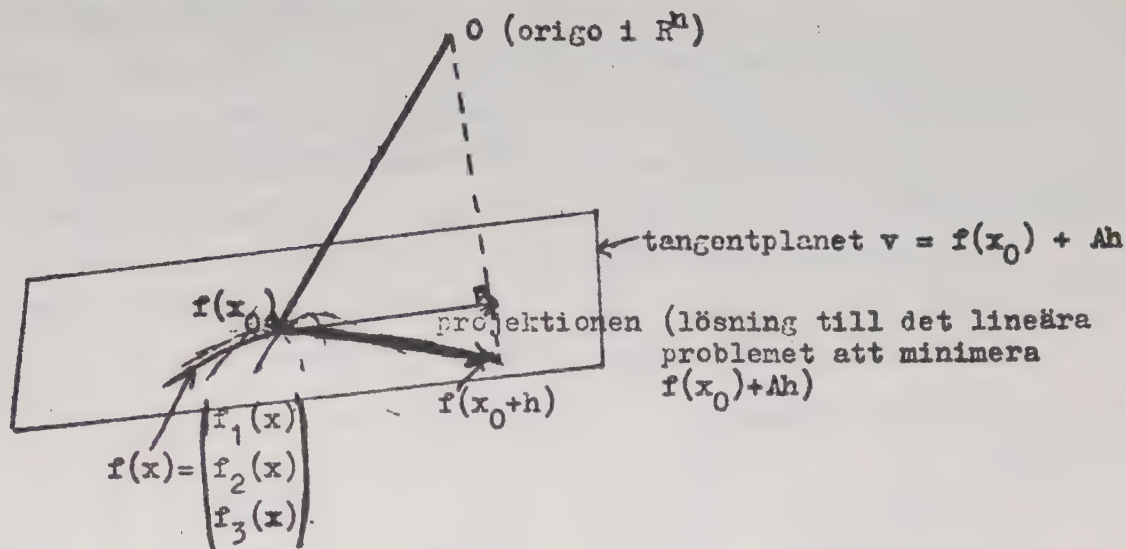


Fig. A1

Projektionen av vektorn från origo i R^n till $f(x_0)$ betecknas $P_{R(A)}f(x_0)$. Punkten närmast origo i tangentplanet blir $f(x_0) - P_{R(A)}f(x_0)$ (beräknas som $f(x_0) - A(A^T A)^{-1} A^T f(x_0)$). Förfarandet upprepas sedan tills avståndet till origo blivit mindre än konstanten ACCU, tills förflyttningen (=längden av projektionen) blivit mindre än ACCU, kvoten mellan längden av projektionen och avståndet till origo blivit mindre än 0.1 eller maximitiden överskridits. Maximitiden i millisekunder är det minsta av värdena MAXT och (antalet punkter som beräknas, dvs n) $\cdot$ (antalet parametrar, dvs m)² $\cdot 2$.

I stället för att alltid gå i projektionens riktning väljs ibland gradientriktningen, dvs en riktning som bestäms direkt av derivatorna till f (finns i matrisen A , riktningen beräknas som $A^T f$).

Om vi inte ibland väljer gradientriktningen som sökriktning så riskerar vi konvergens mot en punkt där A är singular.

Gaussa kan konvergera långsamt, men konvergens mot en stationär punkt, dvs en punkt med $A^T f = 0$ (ekvivalent med $P_{R(A)}f = 0$), är garanterad.

GAUSS-APPROXIMATIONENS TESTFUNKTION

Genom att skriva kommandot TEST (se appendix 1), kan man få utfört ett test, som avgör, om Gauss-approximation eller Simplex-approximation bör användas. Gauss-approximationen används om man är nära en bra lösning av anpassningsproblemet, i annat fall används Simplex-approximation. När testet utförs, används f , x och A på det sätt som definierats i avsnittet Gauss-approximation ovan, dvs $f = (f_1, f_2, \dots, f_n)^T$ är en funktion från R^m till R^n med $f_1 = y_1(1) - y_2(1)$, $y_2(1) =$

$$\sum_{k=1}^r u_k e^{-(\theta_k - t_1)^2 / \sigma_k^2}, \quad x \text{ vektorn med de aktuella parametrarna } (u_1, \theta_1, \dots, \sigma_r)$$

och A matrisen med de partiella derivatorna $a_{ik} = \frac{\partial f_1}{\partial x_k}$ (t.ex. $a_{11} = -e^{-(\theta_1 - t_1)^2 / \sigma_1^2}$)

Vidare beräknas matriser $G^{(j)}$ för $j=1, 2, \dots, n$ med andraderivatorna,

$$\text{så att } g_{ik}^{(j)} = \frac{\partial^2 f_1}{\partial x_i \partial x_k} \quad (\text{t.ex. } g_{11}^{(1)} = 0, g_{12}^{(1)} = g_{21}^{(1)} = \frac{\partial^2 f_1}{\partial u_1 \partial \theta_1} = \frac{2(\theta_1 - t_1)}{\sigma_1^2} e^{-(\theta_1 - t_1)^2 / \sigma_1^2})$$

Därefter beräknas matrisen $A^T A + \sum_{j=1}^n ((I - P_{R(A)}) f)_j G^{(j)}$. Om denna

matris är positivt definit ska Gauss-approximation göras, annars Simplex-approximation. $P_{R(A)}$ beräknas som $A (A^T A)^{-1} A^T$.

Om matrisen är positivt definit avgörs genom Cholesky-uppdelning av matrisen.

GISSNING AV TOPP

Se kommandot GISSA i appendix 1 och DNR i appendix 2.

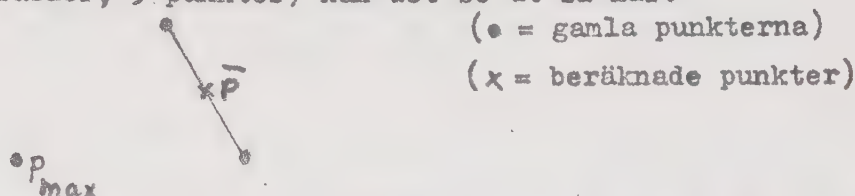
SIMPLEX

Denna metod följer Nelder-Meads beskrivning i A Simplex Method for Function Minimization (1965). Vissa modifieringar har gjorts, vilka i stort sätt följer Parkinson-Hutchinson: An Investigation into the Efficiency of Variants on Simplex Method.

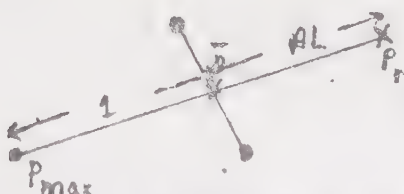
Om vi antar, att vi har mätvärdena $y_1(1), y_1(2), \dots, y_1(n)$ för $t=t_1, t_2, \dots, t_n$, approximerar $y_1(i)$ med $y_2(i) = \sum_{k=1}^r \alpha_k e^{-(\beta_k - t_1)^2 / \gamma_k^2}$ och använder en viktsfunktion rw (som för närvarande är identiskt lika med 1); så betraktar vi $f = (\sum_{i=1}^n rw(i) \cdot (y_1(i) - y_2(i))^2) / \sum_{i=1}^n rw(i)$ som en funktion av $3 \cdot r$ variabler $\alpha_1, \beta_1, \gamma_1, \dots, \alpha_r, \beta_r$ och γ_r . Funktionen f ska minimeras.

Sätt $m=3 \cdot r =$ antalet variabler i f . Idén är, att vi betraktar en startpunkt och m kringliggande punkter $P_0, P_1, \dots, P_m$ i det m -dimensionella rummet R^m , beräknar funktionsvärdena i motsvarande punkter $f_0, f_1, \dots, f_m$, ser efter vilket av dessa funktionsvärden, som är störst, och försöker byta ut motsvarande punkt mot en annan punkt, vars motsvarande funktionsvärde är mindre.

Beteckna det största funktionsvärdet med $f_{\max}$ (motsvarande punkt $P_{\max}$), minsta funktionsvärdet $f_{\min}$ (motsvarande punkt $P_{\min}$) och sätt $\bar{P} =$ mittpunkten bortsett från $P_{\max}$, dvs $\bar{P} = \sum_{k \neq \max} P_k / (m-1)$. Om vi ritar upp fallet $m=2$ (2 variabler, 3 punkter) kan det se ut så här:



Vi söker nu efter ett minsta värde på linjen genom $P_{\max}$ och $\bar{P}$, och börjar med att beräkna $P_r = (1+AL) \cdot \bar{P} - AL \cdot P_{\max}$ och dess funktionsvärde f_r . (AL, BE och GA är beteckningar i programmet på variabler, som i litteraturen ovan betecknats α, β och γ . Beteckningarna AL, BE och GA används för att undvika sammanblandning med α, β och γ i funktionen $\sum_{k=1}^r \alpha_k e^{-(\beta_k - t)^2 / \gamma_k^2}$.)



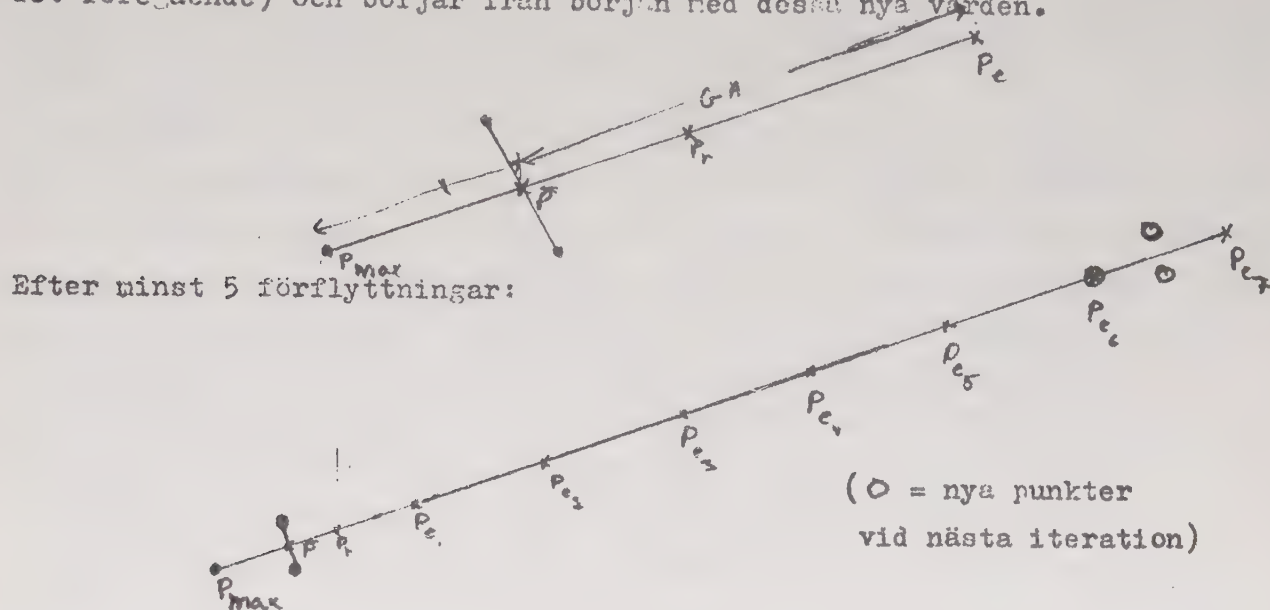
Avståndet mellan P_r och $\bar{P}$ förhåller sig till avståndet mellan $P_{\max}$ och $\bar{P}$ som AL till 1.

Beroende på värdet få f_r skiljer vi nu på fyra olika fall:

1. Om $f_r < f_{\min}$ fortsätter vi i samma riktning och beräknar

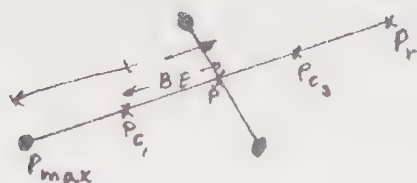
$P_e = GA \cdot P_r + (1 - GA) \cdot \bar{P}$. Om nu $f_e > f_r$ ersätter vi $f_{\max}$ med f_r och fortsätter med dessa nya punkter från början. Om $f_e < f_r$ fortsätter vi i samma riktning med $2 \cdot GA \cdot P_r + (1 - 2 \cdot GA) \cdot \bar{P}$, sedan $3 \cdot GA \cdot P_r + (1 - 3 \cdot GA) \cdot \bar{P}$ o.s.v. så länge funktionsvärdena antas.

Om vi gjort minst 5 förflyttningar i en riktning, flyttar vi samtliga punkter i samma riktning. Vi sätter $P_{\max} = P_e$ (sista värdet mindre än det föregående) och börjar från början med dessa nya värden.



2. Om däremot $f_r \geq f_{\min}$ men det finns minst en punkt förutom $P_{\max}$ som har ett större funktionsvärde f , ersätts $P_{\max}$ med P_r och vi fortsätter med nästa iteration.

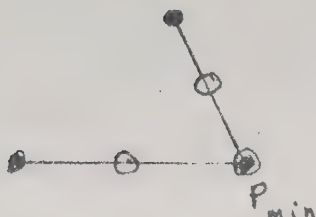
3. Om $f_r < f_{\max}$, men det inte finns något annat $f_i < f_r$, ersätter vi $P_{\max}$ med P_r . I detta fall och i fall 4 ($f_r \geq f_{\max}$) undersöker vi punkter mellan $\bar{P}$ och P_r resp. $P_{\max}$ (se vidare punkt 4).



4. Om $f_r \geq f_{\max}$ beräknas $P_{c_1} = BE \cdot P_{\max} + (1 - BE) \cdot \bar{P}$, om $f_r < f_{\max}$ som

i fall 3 ovan beräknas $P_{c_2} = BE \cdot P_r + (1 - BE) \cdot \bar{P}$. Om motsvarande

funktionsvärde är mindre än $f_{\max}$, ersätts $P_{\max}$, och man börjar om från början igen. I annat fall flyttar man alla punkterna mot den minsta punkten genom att ersätta P_i med $(P_i + P_{\min})/2$ för alla $i \neq \min$ och fortsätter från början igen.



Iterationerna fortsätter tills $f_{\max} - f_{\min} < \text{EPS}$, där EPS är en konstant, dock högst KMAX iterationer.

Som slutpunkt väljs sedan medelvärde av slutpunkterna $P = \sum_{k=0}^m P_k$.

De $m+1$ startpunkterna före första iterationen bestäms på följande sätt:

En startpunkt måste vi själva ge. De övriga m är identiska med startpunktens variabler så när som på en variabel, så att en punkt har ett annat α_1 -värde, en punkt har ett annat β_1 -värde o.s.v. För att t.ex. bestämma punkten med annat α_1 -värde, beräknas funktionsvärdena för startpunkten, startpunkten med α_1 -värdet ersatt med $\alpha_1 + \text{EPS1}$ och startpunkten med α_1 -värdet ersatt med $\alpha_1 - \text{EPS1}$. Genom dessa punkter lägger vi ett andragradspolynom $y = a \cdot t^2 + b \cdot t + c$. Om detta polynom har en positiv andraderivata ($a > 0$), kan vi direkt beräkna, var detta polynom har minimum och sätter det nya α_1 -värdet lika med detta t -värde.

I annat fall ser vi efter, om värdet på $2 \cdot a \cdot t + b$ (förstaderivatan) är skild från 0, och i så fall funktionvärdet för $\alpha_1 + 2 \cdot \text{EPS1}$, om funktionen är avtagande, och funktionsvärdet för $\alpha_1 - 2 \cdot \text{EPS1}$, om funktionen är växande. Genom de senast beräknade 3 punkterna läggs ett nytt andragradspolynom, och man fortsätter beräkningarna som ovan tills man fått $a > 0$. Om första-derivatan är lika med 0, väljs någon av de senast beräknade punkterna (dock så att man får ett α -värde $\neq \alpha_1$). Som steglängd väljs för α -värden EPS1, β -värden EPS2, γ -värden EPS3, δ -värdet EPS4 och ϵ -värdena EPS5 (de två senare endast om Lorentz-kurvor $\sum_{k=1}^r \alpha_k / [1 + \gamma_k \cdot (\beta_k - t)^2 + \delta_k \cdot (\beta_k - t)^4 + \epsilon_k \cdot (\beta_k - t)^6]$ används).

STEPIT

Programmet har utarbetats av J. P. Chandler, Physics Dept., Indiana University, och har erhållits av Quantum Chemistry Program Exchange (QCPE), Indiana University, Bloomington, Indiana, USA. Programmet är beskrivet i programdokumentationen.

Liksom i Simplex-metoden antar vi, att vi har mätvärden $y_1(1), y_1(2), \dots, y_1(n)$ för $t = t_1, t_2, \dots, t_n$, approximerar $y_1(i)$ med

$$y_2(i) = \sum_{k=1}^r \alpha_k e^{-(\beta_k - t)^2 / \gamma_k^2} \text{ och använder en viktsfunktion } rw \text{ (som}$$

för närvarande är identiskt lika med 1), och betraktar

$$f = \left(\sum_{i=1}^n rw(i) \cdot (y_1(i) - y_2(i))^2 \right) / \sum_{i=1}^n rw(i) \text{ som en funktion av } 3 \cdot r (=m)$$

variabler $\alpha_1, \beta_1, \gamma_1, \dots, \alpha_r, \beta_r$ och γ_r . Funktionen f ska minimeras.

Rutinen börjar med en viss rimlighetskontroll av de parametrar, som ska ges till rutinen (t.ex. att man gett rimliga gränser för hur variablerna får variera, och att man gett rimliga start- och slutsteglängder). Steglängden för en parameter k ($k=1, 2, \dots, m$) kallas i programmet $DX(k)$ (initieras med värden $DELTA X(k)$).

I programmet görs ett antal iterationer. Varje iteration kan beskrivas på följande sätt:

Först bestäms för en variabel i taget ett minimum för variabeln, om de andra variablerna inte ändras. Vi börjar med att försöka förbättra värdet på α_1 , och beräknar funktionsvärdena dels för funktionen med de ursprungliga variabelvärdena, dels om α_1 ersätts med $\alpha_1 - DX(1)$, dels om α_1 ersätts med $\alpha_1 + DX(1)$. Om det förstnämnda funktionsvärdet är minst, bestämmer man ett andragradspolynom genom de tre punkterna, bestämmer detta andragradspolynoms minimum och ger α_1 detta värde, om funktionsvärdet för denna punkt är mindre än det ursprungliga funktionsvärdet. Om däremot t.ex. funktionsvärdet för $\alpha_1 + DX(1)$ är minst, beräknas funktionsvärdena för $\alpha_1 + DX(1) \cdot ACK$, $\alpha_1 + DX(1) \cdot ACK^2$, $\alpha_1 + DX(1) \cdot ACK^3$ o.s.v. så länge funktionsvärdena avtar. ACK är en konstant i programmet (för närvarande = 2). När funktionsvärdena börjar öka, läggs ett andragradspolynom genom de tre senast beräknade punkterna, minimum för polynomet bestäms, och funktionsvärdet med $\alpha_1 =$ detta minimivärde beräknas. Därefter behålles det minsta funktionsvärdet, α_1 får motsvarande värdet $\alpha_1 + DX(1) \cdot$ (den framräknade konstanten), och den framräknade konstanten sparas. Med detta α_1 -värde fortsätter man med att minimera β_1 på motsvarande sätt.

När man gått igenom alla variablerna på detta sätt två gånger, övergår man till en annan metod, förutsatt att vissa variabler uppfyller vissa villkor, som inte närmare beskrivs här.

Denna andra metod går ut på, att man samtidigt ändrar samtliga variabler med den steglängd man beräknat ovan. (Om steglängden för en variabel varit större än $DX(i) \cdot ACK$, ändras för övrigt detta $DX(i)$ till $ACK \cdot DX(i)$.) Därefter förflyttar man sig med det ovan beräknade steget $\cdot ACK$, $\text{steget} \cdot ACK^2$, $\text{steget} \cdot ACK^3$ o.s.v. så länge funktionsvärdena avtar, beräknar sedan minimum för andragradspolynomet genom de tre senaste punkterna, samt funktionsvärdet i denna punkt. Punkten med det minsta funktionsvärdet tages därefter till ny startpunkt, och man börjar från början igen med att ändra en variabel i taget.

Beräkningarna fortsätter på detta sätt tills man inte längre åstadkommit någon förbättring efter att man varierat samtliga variabler.

Då testas, om någon steglängd för de enskilda variablerna är större än den minsta steglängd man angett i programmet. Om så inte är fallet sker återhopp från rutinen. I annat fall divideras motsvarande $DX(i)$ med 10. Sedan testas, om man inte kan åstadkomma någon förbättring på grund av att samtliga variabler, som kan varieras, nått de gränser de får röra sig inom, och om så är fallet sker återhopp från rutinen. Slutligen testas, om antalet funktionsberäkningar överstiger $NEMAX$, och i så fall sker återhopp. I annat fall fortsätter man med nästa iteration.

STEPITS FELBERÄKNING

Efter att man gjort en felberäkning med rutinen STEPIT kan man göra en feluppskattning av α -, β - och γ -variablerna genom att skriva ett kommando FEL (se appendix 1). Metoden är närmare beskriven i STEPITS programdokumentation.

I rutinen beräknas först de partiella andraderivatorna med avseende på $\alpha_1, \beta_1, \dots, \gamma_r$ genom att andraderivatorna approximeras med differenser. Man använder då de i rutinen STEPIT senast använda steglängderna 10^{-10} MATRIX-100, där MATRIX är en konstant. Den matris man då får inverteras och multipliceras med 2. Man får då den s.k. felmatrisen.

I denna matris är kvadratroten ur diagonalelementen approximativt standardavvikelsen för felet i motsvarande parameter, dvs roten ur 1-1-elementet e_{11} approximerar felet för α_1 , roten ur e_{22} approximerar felet för β_1 o.s.v. Dessa fel skrivs ut om NTRACE ≥ 0 , och felen dividerade med motsvarande parametrar ($\sqrt{e_{11}}/\alpha_1, \sqrt{e_{22}}/\beta_1$ o.s.v.), dvs relativa felen, sänds till bildskärmen och presenteras på skärmen. För element utanför diagonalen beräknas och utskrivs korrelationerna mellan parametrarna, t.ex. $e_{12}/\sqrt{e_{11} \cdot e_{22}}$ för korrelationen mellan parameter 1 och parameter 2, dvs α_1 och β_1 .

Appendix 5Styrkort för körning av spektralprogrammet vid Lunds Datacentral

Som nämnts i ett tidigare avsnitt är spektralprogrammet uppdelat i två delar, en del för bildskärmsenheten och en del för huvudmaskinen. Om bildskärmsdelen av programmet inte redan finns i bildskärmsenhetens minne, måste den först överföras dit. Man måste då först starta bildskärmen och göra en s.k. bootstrap, som gör, att bildskärmsenheten börjar vänta på överföringen av bildskärmsprogrammet. Hur bootstrap går till är beskrivet på ett papper vid bildskärmen. Man får därefter köra ett program bestående av styrkortet

VRUN XXXXXX,000000,GRAPHIC,S60,20

VASG,AX BE.

VADD BE.SPEKTRAL1557

VFIN

där 000000 skall ersättas med ett projektnummer som tillåter realtidskörningar. Medan programmet överförs blinkar texten ÖVERFÖRING PÅGÅR på skärmen, och när överföringen är klar visas texten SLÅ TILL 16-MS-KLOCKAN. TRYCK PÅ EN TANGENT MEDFÖR HOPP TILL 0301. När man slagit till en strömbrytare för 16-ms-klockan och tryckt på en knapp på tangentbordet, väntar bildskärmen på programmet i huvudmaskinen. Om man vill kan man under denna väntetid ge FX och α -, β - och γ -variablerna de värden man vill ha för första approximationen. Om bildskärmsdelen av programmet redan finns i bildskärmsenheten, får man programmet att vänta på första sändningen från huvudmaskinen genom att skriva kommandot NY KURVA. Då skall texten INPUT FROM 1108 lysa på skärmen. (Om texten OUTPUT TO 1108 lyser, gör först MASTER CLEAR och START med en strömbrytare vid bildskärmen, skriv kommandot STOPPA, gör på nytt MASTER CLEAR och START och skriv kommandot NY KURVA.) När bildskärmsenheten väntar på andra delen av programmet, kan man läsa in en kortbunt av följande utseende:

VRUN XXXXXX,000000,GRAPHIC,S60,50

VADD BE.SPEKTRAL1108

(kort som beskriver den eller de kurvor, som skall analyseras)

VFIN

där 000000 skall ersättas med ett projektnummer, som tillåter realtidskörningar. En kurva beskrivs med följande kort:

kort 1 startvärde för x, dvs x_1 (=TSTART) (i kolumn 1-10 på kortet, med decimalpunkt utskriven (Fortranformat F10.0))
differensen mellan 2 x-värden, dvs $x_2 - x_1$ (=TDIFF) (i kolumn 11-20 på kortet, med decimalpunkt utskriven (format F10.0))

Om annan enhet än kortläsare används skall dess nummer

(enhetsnummer i Fortran) stå i kolumn 30

följande kort: funktionsvärdet på y_k ($k=1, 2, \dots, \text{MAXN}$)

På varje sådant kort stansas i kolumn 1-10 ett värde (format F10.0)

För närvarande kan högst 200 värden läsas in för en kurva.

Om flera kurvor skall analyseras, skall sista y-värdet följas av kortet

VEOF

varefter en ny kurva beskrivs på samma sätt som ovan. En sådan

efterföljande kurva läses in genom kommandot NY INMATNING (se appendix 1).

När inläsningen av en kurva är klar, visas kurvan på bildskärmen, och anpassningen kan börja. När man är klar med alla kurvor, skriver man kommandot STOPPA och stänger av bildskärmen.

Appendix 6

Beskrivning av en testgenerator

För att lättare kunna testa och utvärdera spektralprogrammet har det försetts med ett program, som genererar testdata. Syftet med detta är, att man ska kunna analysera en kurva utan att veta de okända parametrarnas värden från början och sedan i efterhand jämföra det resultat man kommit fram till med den "rätta" lösningen.

För att få lämplig svårighetsgrad på uppgiften och för att få rimliga värden på parametrarna ska man ge testgeneratorn uppgifter om mellan vilka gränser olika parametrar får variera. Mellan gränserna väljs sedan ett tal, så att varje tal fås med lika stor sannolikhet (rektangelfördelat slumpstal mellan minimi- och maximivärdet, om man ska ha ett reellt tal, heltalsdelen av ett slumpstal mellan minimivärdet och maximivärdet+1, om man ska ha ett heltal). Testgeneratorn anropas med styrkortet

VXQT BE.GENERATOR

och efter detta kort ska det komma kort med följande innehåll:

- kort 1 startvärde för x, dvs x_1 (=TSTART)
differensen mellan två x-värden, dvs $x_2 - x_1$ (=TDIFF)
- kort 2 minsta antalet punkter på kurva (=minsta antalet mätvärden)
största antalet punkter på kurvan (=största antalet mätvärden)
- kort 3 minsta antalet toppar
största antalet toppar
- kort 4 sannolikheten för att FX ska ha värdet 1, dvs funktionen
$$\sum_{k=1}^n x_k e^{-(x_k - t)^2} x_k$$
 ska användas (se formel 3.1 - 3.9 på sid 10)
sannolikheten för att FX ska ha värdet 2
sannolikheten för att FX ska ha värdet 3
...
sannolikheten för att FX ska ha värdet 8
- kort 5 sannolikheten för att FX ska ha värdet 9
- kort 6 minsta tillåtna α -värde för topparna
minsta tillåtna Λ -värde för topparna
...
minsta tillåtna ϵ -värde för topparna
- kort 7 största tillåtna α -värde för topparna
...
största tillåtna ϵ -värde för topparna
- kort 8 för eventuellt tillägg av rät linje till kurvan: första x-värdet
 motsvarande minsta funktionsvärde
 motsvarande största funktionsvärde

kort 9 för eventuellt tillägg av rät linje till kurvan: andra x-värdet
 motsvarande minsta funktionsvärde
 motsvarande största funktionsvärde

kort 10 minsta absoluta störningen
 största absoluta störningen

kort 11 minsta relativa störningen
 största relativa störningen

På korten ska första talet stå i kolumn 1 - 10, andra talet i kolumn 11 - 20, tredje talet i kolumn 21 - 30 o.s.v., och talen bör innehålla en decimalpunkt (kortet läses med Fortran-formatet 8F10.5).

På radskrivare utskrivs de inlästa parametervärdena, de framtagna parametervärdena och den beräknade kurvan. Den senare skrivs också ut på en fil, som i programmet benämns fil 1. Denna fil bör vara en skivminnesfil. Om man vill spara detta resultat till en senare körning, bör man före kortet VXQT BE.GENERATOR lägga ett kort

VASG,UP 1.,F2

om man inte tidigare har någon fil med detta namn, respektive kortet VASG,AX 1.

om man redan har en fil med namnet 1, vars gamla innehåll inte ska användas.

Ett exempel: Antal, att vi vill producera en kurva med x-värden från 0 och framåt i steg om 0.1, högst 75 värden (ger största x-värde ≤ 7.4) och minst 50 värden (ger största x-värden ≥ 4.9). Vi vill ha mellan 2 och 4 Lorentz-kurvor + eventuellt en rät linje (motsvarar FX=8 resp. FX=9). Antag, att vi väljer FX=8 med sannolikheten 0.4 och FX=9 med sannolikheten 0.6 (summan av sannolikheterna ska vara lika med 1, varje sannolikhet mellan 0 och 1, om FX får annat värde än mellan 1 och 5 eller 8 och 9 ersätts värdet med 1). Höjden på topparna ska ligga mellan 50 och 100, dvs $50 \leq \alpha\text{-värdena} \leq 100$, läget ska vara mellan $x=1$ och $x=4$, dvs $1 \leq \beta\text{-värdena} \leq 4$, och $\gamma\text{-värdena}$ ska ligga mellan 0.5 och 2. För enkelhet skull bestämmer vi, att alla $\delta\text{-}$ och $\varepsilon\text{-värden}$ ska vara = 0. Om vi lägger till en rät linje, ska den för $x=0$ ha ett y-värde mellan -2 och 2, för $x=5$ ha ett y-värde mellan -1 och 1. Denna rätta linje motsvarar i och för sig en störning, men till denna störning lägger vi en absolut störning mellan -0.1 och 0.1 i varje funktionsvärde och antar vidare att varje funktionsvärde utsätts för en relativ störning mellan -0.001 och 0.001, dvs vi lägger till varje funktionsvärde en störning mellan -0.001 funktionsvärdet och 0.001 funktionsvärdet beräknat ovan med hjälp av $\alpha\text{-}$, $\beta\text{-}$ och $\gamma\text{-värdena}$. Vi antar vidare att vi vill spara resultatet på en fil, så att vi kan använda den vid en senare körning, och att vi inte redan har någon fil med namnet 1.

Den kortbunt vi ska läsa in, kan då få följande utseende:

```
VRUN XXXXXX,000000,GRAPHIC,S60,20
```

```
VASG,AX BE.
```

```
VASG,UP 1.
```

```
VXQT BE.GENERATOR
```

0.0	0.1						
50.0	75.0						
2.0	4.0						
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.4
0.6							
50.0	1.0	0.5	0.0	0.0			
100.0	4.0	2.0	0.0	0.0			
0.0	-2.0	2.0					
5.0	-1.0	1.0					
-0.1	0.1						
-0.001	0.001						

```
VFIN
```

På första kortet utbytes XXXXXX mot det namn man vill ha på körningen och 000000 mot det projektnummer man ska använda.

När man sedan vill köra spektralprogrammet och analysera denna kurva, följer man beskrivningen i appendix 5. Den andra kortbunt (som startar andra delen av spektralprogrammet) får i detta exempel följande utseende:

```
VRUN XXXXXX,000000,GRAPHIC,S60,50
```

```
VASG,AX 1.
```

```
VADD BE.SPEKTRAL1108
```

0.0	0.1	1
-----	-----	---

```
VFIN
```

där XXXXXX och 000000 ersätts med aktuella namn resp. nummer och talet 1 på näst sista kortet står i kolumn 30.

Referenser

1. Chandler, J. P.: Subroutine Stepit, Quantum Chemistry Program Exchange (QCPE), Dept. of Chemistry, Indiana University, Bloomington, Indiana, USA, 1965.
2. Ekenberg, Bertil: Kortfattad beskrivning av bildskärmsutrustningen Univac 1557/1558 vid Lunds Datacentral. Stencil, Institutionen för informationsbehandling, Lund, 1974.
3. Fröberg, Carl-Erik: Lärobok i numerisk analys. Svenska Bokförlaget/Bonniers, Stockholm, 1967.
4. Lederer, C. Michael: Computer Analysis of Spectra. Lawrence Radiation Laboratory, University of California, Berkeley, California.
5. Nelder, J. A., Mead, R.: A Simplex Method for Function Minimization. Computer Journal 7, 1965. Sid 308-313.
6. Parkinson, J. M., Hutchinson, D.: An Investigation into the Efficiency of Variants on the Simplex Method. Ur Lootsma, F. A.(ed): Numerical Methods for Non-linear Optimization. Academic Press, London and New York, 1971. Sid 115-135.
7. Ruhe, Axel: Föreläsningar om Numerisk behandling av matriser. Umeå Universitet, Inst. för Matematik och Statistik, Avd. för Informationsbehandling, Umeå, 1970.
8. Ruhe, Axel, Wedin, Per-Åke: Algorithms for separable nonlinear least squares problem. Report UMINF-47.74, Umeå University, Institute of Mathematics and Statistics, Dept. of Information processing, 1974.
9. Wedin, Per-Åke: On the Gauss-Newton method for the non-linear least squares problem. ITM Arbetsrapport nr 24. Lunds Datacentral, Sölvegatan 18, 223 62 Lund, 1974.
10. Wedin, Per-Åke: The non-linear least squares problem from a numerical point of view. Lund University, Department of Computer Sciences, 1972.
11. Wedin, Per-Åke, Carlsson, L. A.: Gaussa. LDC Program Library, Program Abstract, Lund University Computing Centre, Sweden, 1972.

